

# Multi-Sided Boundary Labeling

Philipp Kindermann\* Benjamin Niedermann† Ignaz Rutter‡ Marcus Schaefer‡  
 André Schulz§ Alexander Wolff†

## Abstract

In the *Boundary Labeling* problem, we are given a set of  $n$  points, referred to as *sites*, inside an axis-parallel rectangle  $R$ , and a set of  $n$  pairwise disjoint rectangular labels that are attached to  $R$  from the outside. The task is to connect the sites to the labels by non-intersecting rectilinear paths, so-called *leaders*, with at most one bend.

In this paper, we study the *Multi-Sided Boundary Labeling* problem, with labels lying on at least two sides of the enclosing rectangle. We present a polynomial-time algorithm that computes a crossing-free leader layout if one exists. So far, such an algorithm has only been known for the cases that labels lie on one side or on two opposite sides of  $R$  (where a crossing-free solution always exists). For the more difficult case where labels lie on adjacent sides, we show how to compute crossing-free leader layouts that maximize the number of labeled points or minimize the total leader length.

## 1 Introduction

Label placement is an important problem in cartography and, more generally, information visualization. Features such as points, lines, and regions in maps, diagrams, and technical drawings often have to be labeled so that users understand better what they see. Even very restricted versions of the label-placement problem are NP-hard [18], which explains why labeling a map manually is a tedious task that has been estimated to take 50% of total map production time [14]. The ACM Computational Geometry Impact Task Force report [6] identified label placement as an important research area. The point-labeling problem in particular has received considerable attention, from practitioners and theoreticians alike. The latter have proposed approximation algorithms for various objectives (label number versus label size), label shapes (such as axis-parallel rectangles or disks), and label-placement models (so-called fixed-position models versus slider models).

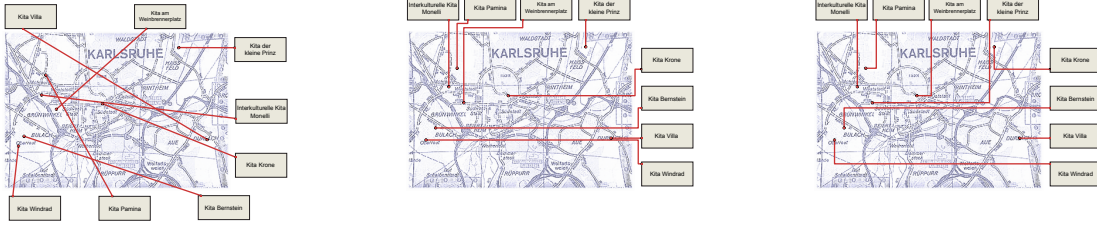
The traditional label-placement models for point labeling insist that a label is placed such that a point on its boundary coincides with the point to be labeled, the *site*. This can make it impossible to label all sites with labels of sufficient size if some sites are very close together. For this reason, Freeman et al. [8] and Zoraster [19] advocated the use of *leaders*, (usually short) line segments that connect sites to labels. In order to make sure that the background image or map remains visible even in the presence of large labels, Bekos et al. [4] took a more radical approach. They introduced models and algorithms for *boundary labeling*, where all labels are placed beyond the boundary of the map and are connected to the sites by straight-line or rectilinear leaders (see Fig. 1).

\*Lehrstuhl für Informatik I, Universität Würzburg, Germany. Email: <http://www1.informatik.uni-wuerzburg.de/en/staff>

†Fakultät für Informatik, Karlsruher Institut für Technologie (KIT), Germany. Email: {rutter,benjamin.niedermann}@kit.edu

‡College of Computing and Digital Media, DePaul University, Chicago, IL, USA. Email: mschaefer@cs.depaul.edu

§Institut für Mathematische Logik und Grundlagenforschung, Universität Münster, Germany. Email: andre.schulz@uni-muenster.de



(a) original labeling of kindergartens in Karlsruhe, Germany (b) *opo*-labeling computed by the algorithm of Bekos et al. [4] (c) *po*-labeling using the same ports as (b)

Fig. 1: A real-world example of boundary labeling with adjacent sides (taken from [4]). For better readability, we have simplified the label texts.

**Problem statement.** Following Bekos et al. [4], we define the BOUNDARY LABELING problem as follows. We are given an axis-parallel rectangle  $R = [0, W] \times [0, H]$ , which is called the *enclosing rectangle*, a set  $P \subset R$  of  $n$  points  $p_1, \dots, p_n$ , called *sites*, within the rectangle  $R$ , and a set  $L$  of  $m \leq n$  axis-parallel rectangles  $\ell_1, \dots, \ell_m$ , called *labels*, that lie in the complement of  $R$  and touch the boundary of  $R$ . No two labels overlap. We denote an instance of the problem by the triplet  $(R, L, P)$ . A *solution* to the problem is a set of  $m$  curves  $c_1, \dots, c_m$ , called *leaders*, that connect sites to labels such that the leaders a) produce a matching between the labels and (a subset of) the sites, b) are contained inside  $R$ , and c) touch the associated labels on the boundary of  $R$ .

A solution is *planar* if the leaders do not intersect. We call an instance *solvable* if a planar solution exists. Note that we do not prescribe which site connects to which label. The endpoint of a curve at a label is called a *port*. We distinguish two versions of the BOUNDARY LABELING problem: either the position of the ports on the boundary of  $R$  is fixed and part of the input, or the ports *slide*, i.e., their exact location is not prescribed.

We restrict our solutions to *po-leaders*, that is, starting at a site, the first line segment of a leader is parallel (*p*) to the side of  $R$  containing the label it leads to, and the second line segment is orthogonal (*o*) to that side; see Fig. 1c. (Fig. 1b shows a labeling with so-called *opo-leaders*, which were investigated by Bekos et al. [4]). Bekos et al. [3, Fig. 12] observed that not every instance (with  $m = n$ ) admits a planar solution with *po-leaders* where all sites are labeled.

**Previous and related work.** For *po*-labeling, Bekos et al. [4] gave a simple quadratic-time algorithm for the one-sided case that, in a first pass, produces a labeling of minimum total leader length by matching sites and ports from bottom to top. In a second pass, their algorithm removes all intersections without increasing the total leader length. This result was improved by Benkert et al. [5] who gave an  $O(n \log n)$ -time algorithm for the same objective function and an  $O(n^3)$ -time algorithm for a very general class of objective functions, including, for example, bend minimization. They extend the latter result to the two-sided case (with labels on opposite sides of  $R$ ), resulting in an  $O(n^8)$ -time algorithm. For the special case of leader-length minimization, Bekos et al. [4] gave a simple dynamic program running in  $O(n^2)$  time. All these algorithms work both for fixed and sliding ports.

Leaders that contain a diagonal part have been studied by Benkert et al. [5] and by Bekos et al. [2]. Recently, Nöllenburg et al. [15] have investigated a dynamic scenario for the one-sided case, Gemsa et al. [9] have used multi-layer boundary labeling to label panorama images, and Fink et al. [7] have boundary labeled focus regions, for example, in interactive on-line maps.

At its core, the boundary label problem asks for a non-intersecting perfect (or maximum) matching on a bipartite graph. Note that an instance may have a planar solution, although all of its leader-length minimal matchings have crossings. In fact, the ratio between a length-minimal solution and a length-minimal crossing-free matching can be arbitrarily bad; see Fig. 2. When connecting points and sites with straight-line segments, the minimum Euclidean matching is necessarily crossing-free. For this case

an  $O(n^{2+\varepsilon})$ -time  $O(n^{1+\varepsilon})$ -space algorithm exists [1]. The minimum-length solution using rectilinear paths with an unbounded number of bends in the presence of obstacles is NP-hard, but there is a 2-approximation [17].

Boundary labeling can also be seen as a graph-drawing problem where the class of graphs to be drawn is restricted to matchings. The restriction concerning the positions of the graph vertices (that is, sites and ports) has been studied for less restricted graph classes under the name *point-set embeddability (PSE)*, usually following the straight-line drawing convention for edges [10]. More recently, PSE has also been combined with the ortho-geodesic drawing convention [12], which generalizes *po*-labeling by allowing edges to make more than one bend. The case where the mapping between ports and sites is given has been studied in VLSI layout [16].

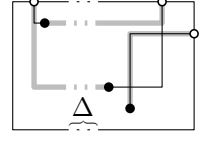


Fig. 2: Length-minimal solutions may have crossings.

**Our contribution.** We investigate the problem TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES where all labels lie on two *adjacent* sides of  $R$ , for example, on the top and right side. Note that point data often comes in a coordinate system; then it is natural to have labels on adjacent sides (for example, opposite the coordinate axes). We argue that this problem is more difficult than the case where labels lie on opposite sides, which has been studied before: with labels on opposite sides, (a) there is always a solution where all sites are labeled (if  $m = n$ ) and (b) a feasible solution can be obtained by considering two instances of the one-sided case.

Our main result is an algorithm that, given an instance with  $n$  labels and  $n$  sites, decides whether a planar solution exists where all sites are labeled and, if yes, computes a layout of the leaders (see Section 3). Our algorithm uses dynamic programming to “guess” a partition of the sites into the two sets that are connected to the leaders on the top side and on the right side. The algorithm runs in  $O(n^2)$  time and uses  $O(n)$  space.

We study several extensions of our main result (see Section 4). First, we show that our approach for fixed ports can also be made to work for sliding ports. Second, we solve the label-number maximization problem (in  $O(n^5)$  time using  $O(n^4)$  space). This is interesting if the position of the sites and labels does not allow for a perfect matching or if there are more sites than labels. Finally, we present a modification of our algorithm that minimizes the leader length (in  $O(n^8 \log n)$  time and  $O(n^6)$  space).

Then in the second part of the paper we investigate the problem THREE-SIDED BOUNDARY LABELING and FOUR-SIDED BOUNDARY LABELING where all labels lie on three or four sides, respectively. To that end we generalize the concept of partitioning the sites labeled by leaders of different sides. In this way we gain sub-instances that we can solve using the algorithm for the two-sided case. All together, we obtain an algorithm solving the four-sided case in  $O(n^{10})$  time and  $O(n)$  space and an algorithm solving the three-sided case in  $O(n^4)$  time and  $O(n)$  space.

**Notation.** We call the labels that lie on the right (top) side of  $R$  *right (top) labels*. The *type* of a label refers to the side of  $R$  on which it is located. The *type* of a leader (or a site) is simply the type of its label. We assume that no two sites lie on the same horizontal or vertical line, and no site lies on a horizontal or vertical line through a port or an edge of a label.

For a solution  $\mathcal{L}$  of a boundary labeling problem, we define several measures that will be used to compare different solutions. We denote the total length of all leaders in  $\mathcal{L}$  by  $\text{length}(\mathcal{L})$ . Moreover, we denote by  $|\mathcal{L}|_x$  the total length of all horizontal segments of leaders that connect a left or right label to a site. Similarly, we denote by  $|\mathcal{L}|_y$  the total length of the vertical segments of leaders that connect top or bottom labels to sites. Note that in general, it is *not* true that  $|\mathcal{L}|_x + |\mathcal{L}|_y = \text{length}(\mathcal{L})$ .

We denote the (uniquely defined) leader connecting a site  $p$  to a port  $t$  of a label  $\ell$  by  $\lambda(p, t)$ . We denote the bend of the leader  $\lambda(p, t)$  by  $\text{bend}(p, t)$ . In the case of fixed ports, we identify ports with labels and simply write  $\lambda(p, \ell)$  and  $\text{bend}(p, \ell)$ , resp.

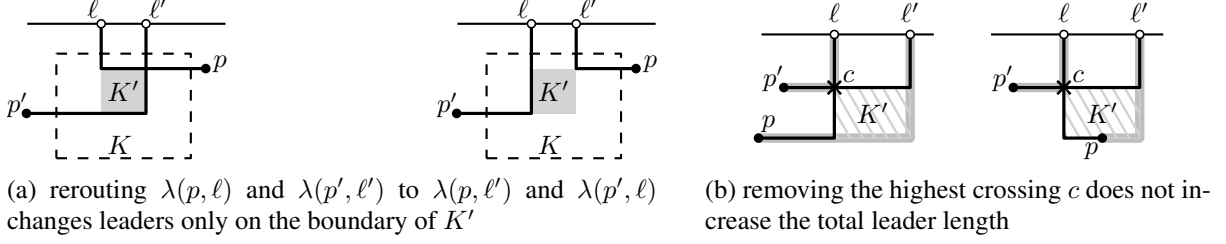


Fig. 3: Illustration of the proof of Lemma 1.

## 2 Structure of Planar Solutions

In this section, we attack our problem presenting a series of structural results of increasing strength. For simplicity, we assume fixed ports. For sliding ports, we can simply fix all ports to the bottom-left corner of their corresponding labels, as described the full version [13]. First we show that we can split a planar two-sided solution into two one-sided solutions by constructing an  $xy$ -monotone, rectilinear curve from the top-right to the bottom-left corner of  $R$ ; see Fig. 4. Afterwards, we provide a necessary and sufficient criterion to decide whether for a given separation there exists a planar solution. This will form the basis of our dynamic programming algorithm, which we present in Section 3.

**Lemma 1.** *Consider a solution  $\mathcal{L}$  for  $(R, L, P)$  and let  $P' \subseteq P$  be sites of the same type. Let  $L' \subseteq L$  be the set of labels of the sites in  $P'$ . Let  $K \subseteq R$  be a rectangle that contains all bends of the leaders of  $P'$ . If the leaders of  $P \setminus P'$  do not intersect  $K$ , then we can rematch  $P'$  and  $L'$  such that the resulting solution  $\mathcal{L}'$  has the following properties: (i) all intersections in  $K$  are removed, (ii) there are no new intersections of leaders outside of  $K$ , (iii)  $|\mathcal{L}'|_x = |\mathcal{L}|_x$ ,  $|\mathcal{L}'|_y = |\mathcal{L}|_y$ , and (iv)  $\text{length}(\mathcal{L}') \leq \text{length}(\mathcal{L})$ .*

*Proof.* Without loss of generality, we assume that  $P'$  contains top sites; the other cases are symmetric. We first prove that, no matter how we change the assignment between  $P'$  and  $L'$ , new intersection points can arise only in  $K$ . Then we show how to establish the claimed solution.

**Claim 1.** *Let  $\ell, \ell' \in L'$  and  $p, p' \in P'$  such that  $\ell$  labels  $p$  and  $\ell'$  labels  $p'$ . Changing the matching by rerouting  $p$  to  $\ell'$  and  $p'$  to  $\ell$  does not introduce new intersections outside of  $K$ .*

Let  $K' \subseteq K$  be the rectangle spanned by  $\text{bend}(p, \ell)$  and  $\text{bend}(p', \ell')$ . When rerouting, we replace  $\lambda(p, \ell) \cup \lambda(p', \ell')$  restricted to the boundary of  $K'$  by its complement with respect to the boundary of  $K'$ ; see Fig. 3a for an example. Thus, any changes concerning the leaders occur only in  $K'$ . The statement of the claim follows.

Since any rematching can be seen as a sequence of pairwise reroutings, the above claim shows that we can rematch  $L'$  and  $P'$  arbitrarily without running the risk of creating new conflicts outside of  $K$ . In order to resolve the conflicts inside  $K$ , we use the length-minimization algorithm for one-sided boundary labeling by Benkert et al. [5], with the sites and ports outside  $K$  projected onto the boundary of  $K$ . Thus, after finitely many such steps, we find a solution  $\mathcal{L}'$  that satisfies properties (i)–(iv).  $\square$

**Definition 1.** *We call an  $xy$ -monotone, rectilinear curve connecting the top-right to the bottom-left corner of  $R$  an  $xy$ -separating curve. We say that a planar solution to TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES is  $xy$ -separated if and only if there exists an  $xy$ -separating curve  $C$  such that*

- a) *the sites that are connected to the top side and all their leaders lie on or above  $C$*
- b) *the sites that are connected to the right side and all their leaders lie below  $C$ .*

It is not hard to see that a planar solution is not  $xy$ -separated if there exists a site  $p$  that is labeled to the right side and a site  $q$  that is labeled to the top side with  $x(p) < x(q)$  and  $y(p) > y(q)$ . There are exactly four patterns in a possible planar solution that satisfy this condition; see Fig. 5. In Lemma 2, we show that these patterns are the only ones that can violate  $xy$ -separability. The proof is given in the full version of the paper [13].

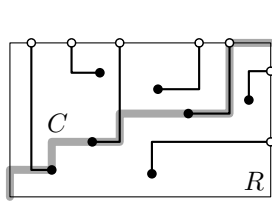


Fig. 4: An  $xy$ -separating curve of a planar solution.

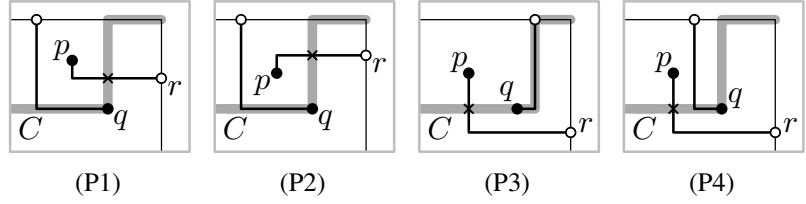


Fig. 5: A planar solution that contains any of the above four patterns P1–P4 is not  $xy$ -separated.

**Lemma 2.** *A planar solution is  $xy$ -separated if and only if it does not contain any of the patterns P1–P4 in Fig. 5.*

*Proof.* Obviously, the planar solution is not  $xy$ -separated if one of these patterns occurs. Let us assume that none of these patterns exists. We construct an  $xy$ -monotone curve  $C$  from the top-right corner of  $R$  to its bottom-left corner. We move to the left whenever possible, and down only when we reach the  $x$ -coordinate of a site  $p$  that is connected to the top, or when we reach the  $x$ -coordinate of a port of a top label, labeling a site  $p$ . If we have to move down, we move down as far as necessary to avoid the corresponding leader, namely down to the  $y$ -coordinate of  $p$ . Finally, when we reach the left boundary of  $R$ , we move down to the bottom-left corner of  $R$ . If  $C$  is free of crossings, then we have found an  $xy$ -separating curve. (For an example, see curve  $C$  in Fig. 4.)

Assume for a contradiction, that a crossing arises during the construction, and consider the topmost such crossing. Note that, by the construction of  $C$ , crossings can only occur with leaders that connect a site  $p$  to a right port  $r$ . We distinguish two cases, based on whether the crossing occurs on a horizontal or a vertical segment of the curve  $C$ .

If  $C$  is crossed on a vertical segment, then this segment belongs to a leader connecting a site  $q$  to a top port  $t$ , and we have reached the  $x$ -coordinate of either the port or the site. Had we, however, reached the  $x$ -coordinate of the port, this would imply a crossing between  $\lambda(p, r)$  and  $\lambda(q, t)$ . Thus, we have reached the  $x$ -coordinate of  $q$ . This means that  $p$  lies to the left of and above  $q$ , and we have found one of the patterns P1 and P2; see Fig. 5.

If  $C$  is crossed on a horizontal segment, then  $p$  must lie above  $r$ . Otherwise, there would be another crossing of  $C$  with the same leader, which is above the current one. This would contradict the choice of the topmost crossing. Consider the previous segment of  $C$ , which is responsible for reaching the  $y$ -coordinate of the current segment. This vertical segment belongs to a leader connecting a site  $q$  to a top port  $t$ . Since leaders do not cross,  $q$  is to the right of  $p$ , and the crossing on  $C$  implies that  $q$  is below  $p$ . We have found one of the patterns P3 and P4; see Fig. 5.  $\square$

Observe that patterns P1 and P2 can be transformed into patterns P4 and P3, respectively, by mirroring the instance diagonally. Next, we prove constructively that, by rerouting pairs of leaders, any planar solution can be transformed into an  $xy$ -separated planar solution.

**Proposition 1.** *If there exists a planar solution  $\mathcal{L}$  to TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES, then there exists an  $xy$ -separated planar solution  $\mathcal{L}'$  with  $\text{length}(\mathcal{L}') \leq \text{length}(\mathcal{L})$ ,  $|\mathcal{L}'|_x \leq |\mathcal{L}|_x$ , and  $|\mathcal{L}'|_y \leq |\mathcal{L}|_y$ .*

*Proof.* Let  $\mathcal{L}$  be a planar solution of minimum total leader length. We show that  $\mathcal{L}$  is  $xy$ -separated. Assume, for the sake of contradiction, that  $\mathcal{L}$  is not  $xy$ -separated. Then, by Lemma 2,  $\mathcal{L}$  contains one of the patterns P1–P4. Without loss of generality, we can assume that the pattern is of type P3 or P4. Otherwise, we mirror the instance diagonally.

Let  $p$  be a right site (with port  $r$ ) and let  $q$  be a top site (with port  $t$ ) such that  $(p, q)$  forms a pattern of type P3 or P4. Among all such patterns, pick one where  $p$  is rightmost. Among all these patterns, pick one where  $q$  is bottommost. Let  $A$  be the rectangle spanned by  $b$  and  $t$ . Let  $B$  be the rectangle spanned

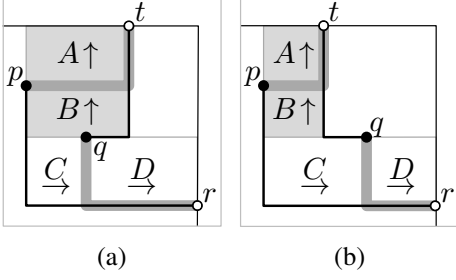


Fig. 6: Types (top =  $\uparrow$  / right =  $\rightarrow$ ) of the sites inside rectangles  $A$ ,  $B$ ,  $C$ , and  $D$ . Left: pattern P3, right: pattern P4.

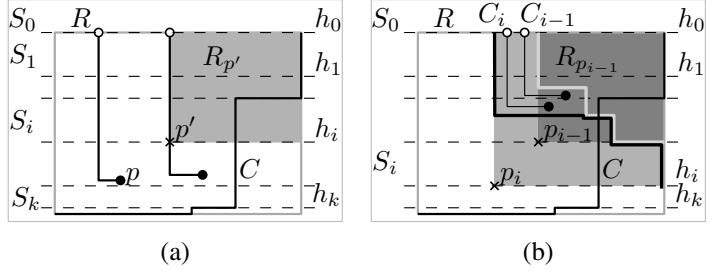


Fig. 7: The strip condition. a) The horizontal segments of  $C$  partition the strips  $S_0, S_1, \dots, S_k$ . b) Constructing a planar labeling from a sequence of valid rectangles.

by  $\text{bend}(q, t)$  and  $p$ . Let  $C$  be the rectangle spanned by  $q$  and  $\text{bend}(p, r)$ . Let  $D$  be the rectangle spanned by  $q$  and  $r$ . By the choice of the pairs  $(p, r)$  and  $(q, t)$ , we have the following properties; see Fig. 6.

- (i) Sites in the interior of  $A$  and  $B$  are connected to the top.
- (ii) Sites in the interiors of  $C$  and  $D$  are connected to the right.

Property (i) is due to the choice of  $p$  as the rightmost site involved in such a pattern. Similarly, property (ii) is due to the choice of  $q$  as the bottommost site that forms a pattern with  $p$ .

Our goal is to change the labeling by rerouting  $p$  to  $t$  and  $q$  to  $r$ , which decreases the total leader length, but may introduce crossings. We then use Lemma 1 to remove the crossings without increasing the total leader length. Let  $\mathcal{L}''$  be the labeling obtained from  $\mathcal{L}$  by rerouting  $p$  to  $t$  and  $q$  to  $r$ . We have  $|\mathcal{L}''|_y \leq |\mathcal{L}|_y - (y(p) - y(q))$  and  $|\mathcal{L}''|_x = |\mathcal{L}|_x - (x(q) - x(p))$ . Moreover,  $\text{length}(\mathcal{L}'') \leq \text{length}(\mathcal{L}) - 2(y(p) - y(q))$ , as at least twice the vertical distance between  $p$  and  $q$  is saved; see Fig. 6. Since the original labeling was planar, crossings may only arise on the horizontal segment of  $\lambda(p, t)$  and on the vertical segment of  $\lambda(q, r)$ .

By properties (i) and (ii) all leaders that cross the new leader  $\lambda(q, r)$  have their bends inside  $C$ , and all leaders that cross the new leader  $\lambda(p, t)$  have their bends inside  $B$ . Thus, we can apply Lemma 1 to the rectangles  $B$  and  $C$  to resolve all new crossings. The resulting planar solution  $\mathcal{L}'$  is planar and has length less than  $\text{length}(\mathcal{L})$ . This is a contradiction to the choice of  $\mathcal{L}$ .  $\square$

Since every solvable instance of TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES admits an  $xy$ -separated planar solution, it suffices to search for such a solution. Moreover, an  $xy$ -separated planar solution that minimizes the total leader length is a solution of minimum length. In Lemma 3 we provide a necessary and sufficient criterion to decide whether, for a given  $xy$ -monotone curve  $C$ , there is a planar solution that is separated by  $C$ . We denote the region of  $R$  above  $C$  by  $R_T$  and the region of  $R$  below  $C$  by  $R_R$ . These regions are relatively open at  $C$ . For each horizontal segment of  $C$  consider the horizontal line through the segment. We denote the part of these lines within  $R$  by  $h_1, \dots, h_k$ , respectively. Further, let  $h_0$  be the top edge of  $R$ . The line segments  $h_1, \dots, h_k$  partition  $R_T$  into  $k$  strips, which we denote by  $S_1, \dots, S_k$  from top to bottom, such that each strip  $S_i$  is bounded by  $h_i$  from below for  $i = 1, \dots, k$ ; see Fig. 7a. Additionally, we define  $S_0$  to be the empty strip that coincides with  $h_0$ . Note that this strip cannot contain any site of  $P$ . For any point  $p$  on one of the horizontal lines  $h_i$ , we define the rectangle  $R_p$ , spanned by the top-right corner of  $R$  and  $p$ . We define  $R_p$  such that it is closed but does *not* contain its top-left corner. In particular, we consider the port of a top label as contained in  $R_p$ , only if it is not the upper left corner.

A rectangle  $R_p$  is *valid* if the number of sites of  $P$  above  $C$  that belong to  $R_p$  is at least as large as the number of ports on the top side of  $R_p$ . The central idea is that the sites of  $P$  inside a valid rectangle  $R_p$  can be connected to labels on the top side of the valid rectangle by leaders that are completely contained inside the rectangle. We are now ready to present the *strip condition*.

**Condition 1.** The strip condition of strip  $S_i$  is satisfied if there exists a point  $p_i \in h_i \cap R_T$  such that  $R_{p_i}$

is valid.

We now prove that, for a given  $xy$ -monotone curve  $C$  going from the top-right corner to the bottom-left corner of  $R$ , there exists a planar solution in  $R_T$  for the top labels if and only if  $C$  satisfies the strip condition for all strips  $S_0, \dots, S_k$  in  $R_T$ . For the proof we call a region  $S \subseteq R$  *balanced* if it contains the same number of sites as it contains ports.

**Lemma 3.** *Let  $C$  be an  $xy$ -monotone curve from the top-right corner of  $R$  to the bottom-left corner of  $R$ . Let  $P' \subseteq P$  be the sites that are in  $R_T$ . There is a planar solution that uses all top labels of  $R$  to label sites in  $P'$  in such a way that all leaders are in  $R_T$  if and only if each horizontal strip  $S_i$ , as defined above, satisfies the strip condition.*

*Proof.* To show that the conditions are necessary, let  $\mathcal{L}$  be a planar solution for which all top leaders are above  $C$ . Consider strip  $S_i$ , which is bounded from below by line  $h_i$ ,  $0 \leq i \leq k$ . If there is no site of  $P'$  below  $h_i$ , rectangle  $R_p$  is clearly valid, where  $p$  is the intersection of  $h_i$  with the left side of  $R$ , and thus the strip condition is satisfied. Hence, assume that there is a site  $p \in P'$  that is labeled by a top label, and is in strip  $S_j$  with  $j > i$ ; see Fig. 7a. Then, the vertical segment of this leader crosses  $h_i$  in  $R_T$ . Let  $p'$  denote the rightmost such crossing of a leader of a site in  $P'$  with  $h_i$ . We claim that  $R_{p'}$  is valid. To see this, observe that all sites of  $P'$  top-right of  $p'$  are contained in  $R_{p'}$ . Since no leader may cross the vertical segments defining  $p'$ , the number of sites in  $R_{p'} \cap R_T$  is balanced, i.e.,  $R_{p'}$  is valid.

Conversely, we show that if the conditions are satisfied, then a corresponding planar solution exists. Let  $S_k$  be the last strip that contains sites of  $P'$ . For  $i = 0, \dots, k-1$ , let  $p'_i$  denote the rightmost point of  $h_i \cap R_T$  such that  $R_{p'_i}$  is valid. We define  $p_i$  to be the point on  $h_i \cap R_T$ , whose  $x$ -coordinate is  $\min_{j \leq i} \{x(p'_j)\}$ . Note that  $R_{p_i}$  is a valid rectangle, as, by definition, it completely contains some valid rectangle  $R_{p'_j}$  with  $x(p'_j) = x(p_i)$ . Also by definition the sequence formed by the points  $p_i$  has decreasing  $x$ -coordinates, i.e., the  $R_{p_i}$  grow to the left; see Fig. 7b.

We prove inductively that, for each  $i = 0, \dots, k$ , there is a planar labeling  $\mathcal{L}_i$  that matches the labels on the top side of  $R_{p_i}$  to points contained in  $R_{p_i}$ , in such a way that there exists an  $xy$ -monotone curve  $C_i$  from the top-left to the bottom-right corner of  $R_{p_i}$  that separates the labeled sites from the unlabeled sites without intersecting any leaders. Then  $\mathcal{L}_k$  is the claimed labeling.

For  $i = 0$ ,  $\mathcal{L}_0 = \emptyset$  is a planar solution. Consider a strip  $S_i$  with  $0 < i \leq k$ ; see Fig. 7b. By the induction hypothesis, we have a curve  $C_{i-1}$  and a planar labeling  $\mathcal{L}_{i-1}$ , which matches the labels on the top side of  $R_{p_{i-1}}$  to the sites in  $R_{p_{i-1}}$  above  $C_{i-1}$ . To extend it to a planar solution  $\mathcal{L}_i$ , we additionally need to match the remaining labels on the top side of  $R_{p_i}$  and construct a corresponding curve  $C_i$ . Let  $P_i$  denote the set of unlabeled sites in  $R_{p_i}$ . By the validity of  $R_{p_i}$ , this number is at least as large as the number of unused ports at the top side of  $R_{p_i}$ . We match these ports from top to bottom to the topmost sites of  $P_i$ ; the result is the claimed planar labeling  $\mathcal{L}_i$ . The ordering ensures that no two of the new leaders cross. Moreover, no leader crosses the curve  $C_{i-1}$ , and hence such leaders cannot cross leaders in  $\mathcal{L}_{i-1}$ . It remains to construct the curve  $C_i$ . For this, we start at the top-left corner of  $R_{p_i}$  and move down vertically, until we have passed all labeled sites. We then move right until we either hit  $C_{i-1}$  or the right side of  $R$ . In the former case, we follow  $C_{i-1}$  until we arrive at the right side of  $R$ . Finally, we move down until we arrive at the bottom-right corner of  $R_{p_i}$ . Note that all labeled sites are above  $C_i$ , unlabeled sites are below  $C_i$ , and no leader is crossed by  $C_i$ . This is true since we first move below the new leaders and then follow the previous curve  $C_{i-1}$ .  $\square$

A symmetric strip condition (with vertical strips) can be obtained for the right region  $R_R$  of a partitioned instance. The characterization is completely symmetric.

### 3 The Algorithm

Now we describe how to find an  $xy$ -monotone chain  $C$  that satisfies the strip conditions. For that purpose we only consider  $xy$ -monotone chains that lie on the dual of the grid induced by the sites and ports of

the given instance. When traversing this grid from grid point to grid point, we either pass a site (*site event*) or a port (*port event*). By passing a site, we decide if the site is connected to the top or to the right side. Clearly, there is an exponential number of possible  $xy$ -monotone traversals through the grid. In the following, we describe a dynamic program that finds an  $xy$ -separating chain in  $O(n^3)$  time.

Let there be  $m_R$  ports on the right side of  $R$  and  $m_T$  ports on the top side of  $R$ , then the grid has size  $[n + m_T + 1] \times [n + m_R + 1]$ . We define the grid points as  $G(x, y)$ ,  $0 \leq x \leq n + m_T + 1$ ,  $0 \leq y \leq n + m_R + 1$  with  $G(0, 0)$  being the bottom-left and  $r := G(n + m_T + 1, n + m_R + 1)$  being the top-right corner of  $R$ . Further, we define  $G_x(s) := x(G(s, 0))$  and  $G_y(t) := y(G(0, t))$ .

An entry in the table of our dynamic program is described by three values. The first two values are  $s$  and  $t$ , which give the position of the current search for the curve  $C$ . The interpretation is that the entry encodes the possible  $xy$ -monotone curves from  $r$  to  $p_C := G(s, t)$ ; see Fig. 8. The remaining value  $u$  denotes the number of sites above  $C$  in the rectangle spanned by  $r$  and  $p_C$ . Note that it suffices to store  $u$ , as the number of sites below the curve  $C$  can directly be derived from  $u$  and all sites that are contained in the rectangle spanned by  $r$  and  $p_C$ . We denote the first values describing the positions of the curves by the vector  $\mathbf{c} = (s, t)$ . Our goal is to compute a table  $T[\mathbf{c}, u]$  such that  $T[\mathbf{c}, u] = \text{true}$  if and only if there exists an  $xy$ -monotone chain  $C$  such that the following conditions hold. (i) Curve  $C$  starts at  $r$  and ends at  $p_C$ . (ii) Inside the rectangle spanned by  $r$  and  $p_C$ , there are  $u$  sites of  $P$  above  $C$ . (iii) For each strip in the two regions  $R_T$  and  $R_R$  defined by  $C$  the strip condition holds.

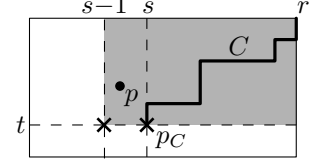


Fig. 8: Possible step of the dynamic program, where  $p$  enters the rectangle spanned by  $r$  and  $G(s - 1, t)$ .

It follows from these conditions, Proposition 1 and Lemma 3 that the instance admits a planar solution if and only if  $T[(0, 0), u] = \text{true}$ , for some  $u$ . Let us now proceed to describe how to compute the table. Initially, we set  $\mathbf{c} = (n + m_T + 1, n + m_R + 1)$ . We initialize the first entry  $T[\mathbf{c}, 0] = \text{true}$ . The remaining entries are initialized with *false*.

Let  $\mathbf{c} := (s, t)$  be the current grid point we checked as endpoint for  $C$ . Based on the table  $T[\mathbf{c}, \cdot]$  we then compute the entries  $T[\mathbf{c} - \Delta\mathbf{c}, \cdot]$  where the vector  $\Delta\mathbf{c} = (\Delta s, \Delta t)$  is either  $(0, 1)$  or  $(1, 0)$ . We classify such steps, depending on whether we cross a site or a port. We give a full description for  $\Delta\mathbf{c} = (1, 0)$ , i.e., we decrease  $s$  by 1. The other case is completely symmetric. Assume  $T[\mathbf{c}, u] = \text{true}$ . We distinguish two cases, based on whether we cross a site or a port.

**Case 1:** Going from  $s$  to  $s - 1$  is a site event, i.e., there is a site  $p$  with  $G_x(s) > x(p) > G_x(s - 1)$ . Note that by our assumption of general position and the definition of the coordinates, the site  $p$  is unique. If  $y(p) > G_y(t)$ , then  $p$  enters the rectangle spanned by  $G(s - 1, t)$  and  $r$ , and it is located above  $C$ ; see Fig. 8. We thus set  $T[\mathbf{c} - \Delta\mathbf{c}, u + 1] = \text{true}$ . Otherwise we set  $T[\mathbf{c} - \Delta\mathbf{c}, u] = \text{true}$ . Note that the strip conditions remain satisfied since we do not decrease the number of sites in any region.

**Case 2:** Going from  $s$  to  $s - 1$  is a port event, i.e., there is a label  $\ell$  on the top side, whose port is between  $G_x(s - 1)$  and  $G_x(s)$ . Thus, the region above  $C$  contains one more label. We therefore check the strip condition for the strip above the horizontal line through  $G(s - 1, t)$ . If it is satisfied, we set  $T[\mathbf{c} - \Delta\mathbf{c}, u] = \text{true}$ .

If  $T[\mathbf{c}, u] = \text{false}$ , there is no  $xy$ -monotone chain that satisfies the conditions given above, so the it suffices to only look at the *true* table entries. This immediately gives us a polynomial-time algorithm for TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES. The running time crucially relies on the number of strip conditions that need to be checked. We show that after a  $O(n^2)$  preprocessing phase, such queries can be answered in  $O(1)$  time.

To implement the test of the strip conditions, we use a table  $B_T$ , which stores in the position  $B_T[s, t]$  how large a deficit of sites to the right can be compensated by sites above and to the left of  $G(s, t)$ . That is,  $B_T[s, t]$  is the maximum value  $k$  such that there exists a rectangle  $K_{B_T[s, t]}$  with lower right corner  $G(s, t)$  whose top side is bounded by the top side of  $R$ , and that contains  $k$  more sites in its interior, than it has ports on its top side. Once we have computed this matrix, it is possible to query



the strip condition in the dynamic program that computes  $T$  in  $O(1)$  time. Namely, assume we have an entry  $T[(s, t), u]$ , and we wish to check its strip condition. Consider a curve  $C$  from  $r$  to  $G(s, t)$  such that  $u$  sites are above  $C$ . The strip condition is satisfied if and only if  $u + B_T[s, t]$  is at least as large as the number of top ports to the right of  $G(s, t)$ . Namely, the rectangle spanned by the lower left corner of  $K_{B_T[s, t]}$  and  $r$  then contains at least  $u + B_T[s, t]$  sites, which is an upper bound on the number of ports on the top side of that rectangle.

We now show how to compute  $B_T$  in  $O(n^2)$  time. We compute each row separately, starting from the left side. We initialize  $B_T[0, t] = 0$  for  $t = 0, \dots, n + m_R + 2$ , as in the final column, no deficit can be compensated. The matrix  $B$  can be filled by a horizontal sweep. Namely, consider the entry  $B_T[s, t]$ , and assume we have already computed  $B_T[s - 1, t]$ . If the step from  $s - 1$  to  $s$  is a site event, this increases the amount of deficit we can compensate by 1. Otherwise, it is a port event. This decreases the amount of deficit we can compensate by 1. However, the compensation potential never goes below 0. We obtain

$$B_T[s, t] = \begin{cases} B_T[s - 1, t] + 1 & \text{if step is site event,} \\ \max\{B_T[s - 1, t] - 1, 0\} & \text{if step is port event.} \end{cases}$$

The table can be clearly filled out in  $O(n^2)$  time. A similar matrix  $B_R$  can be computed for the vertical strips. Altogether, this yields an algorithm for TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES that runs in  $O(n^3)$  time and uses  $O(n^3)$  space. However, the entries of each row and column of  $T$  depend only on the previous row and column, which allows us to reduce the storage requirement to  $O(n^2)$ . Using Hirschberg's algorithm [11], we can still backtrack the dynamic program and find a solution corresponding to an entry in the last cell in the same running time. We have the following theorem.

**Theorem 1.** TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES *can be solved in  $O(n^3)$  time using  $O(n^2)$  space.*

In order to increase the performance of our algorithm, we can reduce the number of dimensions of the table  $T$  by 1. As a first step, we show that for any search position  $\mathbf{c}$ , the possible values of  $u$ , for which  $T[\mathbf{c}, u] = \text{true}$  form an interval.

Our next goal is to improve the performance of our algorithm. As a first step, we show that for any search position  $\mathbf{c}$ , the possible values of  $u$ , for which  $T[\mathbf{c}, u] = \text{true}$  form an interval.

**Lemma 4.** *Let  $T[\mathbf{c}, u] = T[\mathbf{c}, u'] = \text{true}$  with  $u \leq u'$ . Then  $T[\mathbf{c}, u''] = \text{true}$  for  $u \leq u'' \leq u'$ .*

*Proof.* Let  $C$  be the curve corresponding to the entry  $T[\mathbf{c}, u]$ . That is  $C$  connects  $r$  to  $p_C$  such that  $u$  sites in the rectangle spanned by  $p_C$  and  $r$  are above  $C$ , and the strip conditions (both above and below  $C$ ) are satisfied. Similarly, let  $C'$  be the curve corresponding to  $T[\mathbf{c}, u']$ .

Since  $u$  and  $u'$  differ, there is a rightmost site  $p$ , such that  $p$  is below  $C$  and above  $C'$ . Let  $v$  and  $v'$  be the grid points of  $C$  and  $C'$  that are immediately to the left of  $p$ . Note that  $v$  is above  $v'$  since  $C$  is above  $p$  and  $C'$  is below it. Consider the  $C''$ , which starts at  $r$  and follows  $C$  until  $v$ , then it moves down vertically to  $v'$ , and from there follows  $C'$  to  $p$ . Obviously  $C''$  is an  $xy$ -monotone curve, and it has above it the same sites as  $C'$ , except for  $p$ , which is below it. Thus there are  $u'' = u' - 1$  sites above  $C''$  in the rectangle spanned by  $p$  and  $r$ . If all strips defined by  $C''$  satisfy the strip condition, then this implies  $T[\mathbf{c}, u''] = \text{true}$ .

To see that the strip conditions are indeed satisfied, consider a horizontal strip  $S''$  defined by  $C''$ . Let  $S$  be the lowest horizontal strip defined by  $C$  that is not below the lower boundary of  $S''$ . We know that  $S$  fulfills the strip condition, which is witnessed by some valid rectangle  $K$ . We can enlarge this rectangle vertically such that it touches the lower boundary of  $S''$ . The enlarged rectangle contains at least as many sites above  $C''$  as there were above  $C$  in  $K$ . Hence it is a valid rectangle and the strip condition for  $S''$  holds. An analogous statement holds for the vertical strips since  $C''$  is above  $C'$  at every  $x$ -coordinate.  $\square$

Using Lemma 4, we can reduce the dimension of the table  $T$  by 1. It suffices to store at each entry  $T[c]$  the boundaries of the  $u$ -interval. This reduces the amount of storage to  $O(n^2)$  without increasing the running time. Using Hirschberg's algorithm, the storage for  $T$  even decreases to  $O(n)$ . However, tables  $B_T$  and  $B_R$  have size  $O(n^2)$ . Next, we reduce the running time to  $O(n^2)$ . An entry in  $B_T[s, t]$  tells us which deficits can be compensated. This can also be interpreted as a lower bound on the number of sites a curve from  $r$  to  $G(s, t)$  must have above it, in order to satisfy the horizontal strip condition. Namely, let  $\tau_{s,t}$  denote the number of ports on the top side of the rectangle spanned by  $G(s, t)$  and  $r$ . Then  $u \geq \tau_{s,t} - B_T[s, t]$  is equivalent to satisfying the horizontal strip condition for the strip directly above  $G(s, t)$ . Similarly, the corresponding entry  $B_R[s, t]$  gives a lower bound on the number of sites below such a curve, which in turn, together with the number of sites contained in the rectangle spanned by  $G(s, t)$  and  $r$  implies an *upper bound* on the number of sites above the curve. Thus,  $B_T$ ,  $B_R$ , and the information how many sites, top ports and right ports are in the rectangle spanned by  $G(s, t)$  and  $r$  together imply a lower and an upper bound, and thus an interval of  $u$ -values, for which the horizontal and vertical strip conditions at  $G(s, t)$  is satisfied. Thus, in the program we rather just intersect this interval with the union of the intervals obtained from  $T[(s, t) - \Delta c]$ , where  $\Delta c$  has exactly one non-zero entry, which is 1. Consequently, the amount of work per entry of  $T$  is still  $O(1)$ .

Next, we would like to reduce the storage using Hirschberg's algorithm [11], which immediately reduces the storage requirement of  $T$  to  $O(n)$ . We would like to reduce the storage requirement for  $B_T$  and  $B_R$  by using Hirschberg's algorithm as well. However,  $B_T$  is computed from left to right and  $B_R$  is computed from bottom to top, which is exactly opposite to  $T$ , which is computed from top-right to bottom-left. The main idea is to let the dynamic programs for computing  $B_T$  and  $B_R$  run backwards, by precomputing the entries of  $B_T$  and  $B_R$  on the top and right side, and then running the updates backwards. Then Hirschberg's algorithm can be used and the algorithms can run in a synchronized manner, such that at each point in time the required data is available, using only  $O(n)$  space. However, the update  $B_T[s, t] = \max\{B_T[s-1, t] - 1, 0\}$  is not easily reversible. Namely, when running the dynamic program backwards it is not clear, whether  $B_T[s, t] = 0$  implies  $B_T[s-1, t] = 0$  or  $B_T[s-1, t] = 1$ .

To remedy this problem, consider a fixed column  $s$  of the table corresponding to a port event, and consider the circumstances under which  $B_T[s-1, t] - 1 = -1$ , i.e.,  $B_T[s-1, t] = 0$ . This implies that, for any rectangle  $K$  with lower right corner  $G(s-1, t)$  whose top side is contained in the top side of  $R$ , there are at most as many sites in  $K$  as there are ports in the top side of  $K$ . Assume that this is the case for some fixed value  $t_0$ , i.e.  $B_T[s-1, t_0]$ . Since the possible rectangles for an entry  $B_T[s-1, t]$  with  $t \geq t_0$  contain at most as many sites as the ones for  $B_T[s-1, t_0]$ , this implies  $B_T[s-1, t_0] = B_T[s-1, t] = 0$  for all  $t \geq t_0$ . If on the other hand,  $t_0$  is such that  $B_T[s-1, t_0] > 0$ , then the rectangles corresponding to  $B_T[s-1, t]$  for  $t < t_0$  contain at least as many sites as the ones for  $B_T[s-1, t_0]$ , and we have  $B_T[s-1, t] \geq B_T[s-1, t_0]$  for  $t < t_0$ . Thus, there is a single gap  $t_0$ , such that for all  $t \geq t_0$  we have  $B_T[s-1, t] = 0$  and for  $t < t_0$ , we have  $B_T[s-1, t] > 0$ ; see Fig. 9. Storing this gap for each column  $s$  that is a port event allows to efficiently reverse the dynamic program. Note that storing one value per column only incurs  $O(n)$  space overhead. Of course, the same approach works for the dynamic program computing  $B_R$ . We have shown the following theorem.

**Theorem 2.** TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES *can be solved in  $O(n^2)$  time using  $O(n)$  space.*

## 4 Extensions

The techniques we used to obtain Theorem 2 can be applied to solve a variety of different extension of the two-sided labeling problem with adjacent sides. We now show how to a) generalize to sliding ports instead of fixed ports, b) maximize the number of labeled sites, and c) minimize the total leader length in a planar solution.

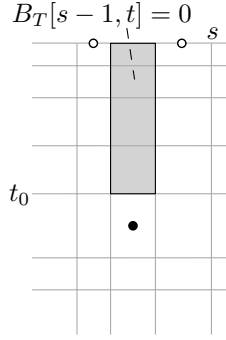


Fig. 9: The gap  $t_0$  is defined such that, for any  $t \geq t_0$ , we have  $B_T[s-1, t] = 0$  and, for any  $t < t_0$ , we have  $B_T[s-1, t] > 0$ .

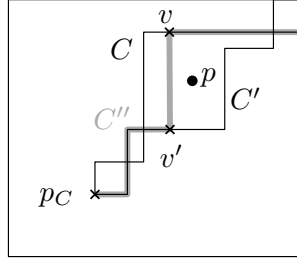


Fig. 10: Sketch for the proof of Lemma 4.

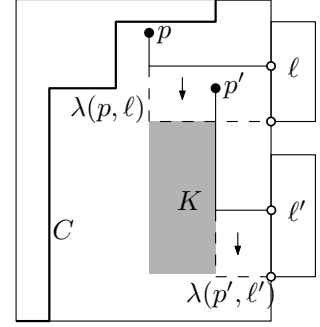


Fig. 11: Sketch for the proof of Lemma 5

## 4.1 Sliding Ports

First, observe that Proposition 1, which guarantees the existence of an  $xy$ -separated planar solution, also holds for sliding ports. The same proofs apply by conceptually fixing the ports of a given planar when applying the rerouting operations. The following lemma shows that, without loss of generality, we can simply fix all ports to the bottom-left corner of their corresponding labels. This immediately solves the problem.

**Lemma 5.** *If there exists an  $xy$ -separated planar solution  $\mathcal{L}$  for the two-sided boundary labeling problem with adjacent sides and sliding ports, then there also exists an  $xy$ -separated planar solution  $\mathcal{L}'$  in which the ports are fixed to the bottom left corners of the labels.*

*Proof.* We show how to transform  $\mathcal{L}$  into  $\mathcal{L}'$ . Let  $C$  be the  $xy$ -monotone curve that separates the top leaders from the right leaders of  $\mathcal{L}$ . We move the ports induced by  $\mathcal{L}$  to the bottom-left corner of their corresponding labels such that the assignment between labels and sites remains; see Fig.11. Obviously, the bends of the leaders connected to the right side only move downwards. Thus, the leaders remain below  $C$ . Symmetrically, the bends of the leaders connected to the top side only move to the left and thus these leaders remain above  $C$ .

Consequently, only conflicts between the same type of leaders can arise. Consider the topmost intersection of two leaders  $\lambda(p, \ell)$  and  $\lambda(p', \ell')$  connected to the right side and assume that  $p$  lies left of  $p'$ . Let  $K$  be the rectangle that is spanned by the bends of  $\lambda(p, \ell)$  and  $\lambda(p', \ell')$ . Due to moving the ports downwards, the leaders remain below  $C$  and the bend of  $\lambda(p', \ell')$  must lie below  $\lambda(p, \ell)$ . Consequently,  $K$  lies completely in  $R_R$ . In order to resolve the conflict, we reroute  $p$  to  $\ell'$  and  $p'$  to  $\ell$  using the bottom-left corners of  $\ell$  and  $\ell'$  as ports. Obviously, the leaders only change on  $\partial K$ . Hence, new conflicts can only arise on the left and bottom sides of  $K$ . In particular, only the leader of  $\ell'$  can be involved in new conflicts, while the leader of  $\ell$  is free of any conflict. Thus, after finitely many such steps we have resolved all conflicts, from top to bottom. Symmetric arguments apply for the leaders connected to the top side.  $\square$

**Theorem 3.** TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES AND SLIDING PORTS can be solved in  $O(n^2)$  time using  $O(n)$  space.

## 4.2 Maximizing the Number of Labeled Sites

So far our algorithm only returns a leader layout if there is a planar solution that matches sites and labels perfectly. It is, however, of great use to also have an algorithm that maximizes the number of labels connected to sites in a planar solution. We can achieve this by removing labels from a given instance and using our algorithm to decide whether a crossing-free solution exists.

Lemma 5 shows that we can move top ports to the left and right ports to the bottom without making a solvable instance unsolvable. Thus, it suffices to remove the right-most top labels and the top-most right labels. Let  $k$  be the number of labels we want to use. Further, let  $k_T$  be the number of top labels and  $k_R$  be the number of right labels we want to use. Obviously,  $k_T + k_R = k$ . For a given  $k$ , we can decide whether a crossing-free solution that uses exactly  $k$  labels exists by removing the  $m_T - k_T$  right-most top labels and the  $m_R - k_R$  top-most right labels for any possible  $k_T$  and  $k_R$ . We therefore start with  $k_T = \min\{k, m_T\}$  and  $k_R = k - k_T$ . We keep decreasing  $k_T$  and increasing  $k_R$  by 1, until a crossing-free solution is found or  $k_R = \min\{k, m_R\}$ . In the latter case, no crossing-free solution that uses exactly  $k$  labels exists. With this approach we can use binary search to find the maximum  $k$ , using our algorithm up to  $k$  times per step. Since  $k \leq n$ , this yields an algorithm for TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES that maximizes the number of labeled sites that runs in  $O(n^3 \log n)$  time and uses  $O(n)$  space.

**Theorem 4.** TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES can be solved in  $O(n^3 \log n)$  time using  $O(n)$  space such that the number of labeled sites is maximized.

### 4.3 Minimizing the total leader length

In order to obtain a length-minimal planar solution, we mainly change the table  $T$  used by the dynamic program. Let  $\mathbf{F}_T = (x_T, y_T)$  be the bottom-left point of the top front for a given  $xy$ -monotone chain  $C$  that starts at  $r$  and ends at  $G(s, t)$ . Similarly, let  $\mathbf{F}_R = (x_R, y_R)$  be the right front for  $C$ . We define  $T[\mathbf{c} = (s, t), u, \mathbf{F}_T, \mathbf{F}_R] = (l, g_T, g_R)$  if there exists an  $xy$ -monotone chain  $C$  such that the following conditions hold, otherwise it contains  $(-1, 0, 0)$ .

- (i) Curve  $C$  starts at the top-right corner  $r$  of  $R$  and ends at  $G(s, t)$ .
- (ii) Inside the rectangle  $K$  spanned by  $r$  and  $G(s, t)$ , there are  $u$  sites of  $P$  above  $C$ .
- (iii) For each strip in the two regions  $R_T$  and  $R_R$  defined by  $C$  the strip condition holds.
- (iv) The sites in  $K \cup F_T \cup F_R$  are connected to the ports on the border of  $K \cup F_T \cup F_R$  such that the arising solution is planar, length-minimal, the sites above  $C$  or in  $F_T$  are only connected to top ports, and the sites below  $C$  or in  $F_R$  are only connected to right ports.
- (v) There are  $g_T$  unlabeled top sites and  $g_R$  unlabeled right sites in  $K$ .

Note that  $g_T$  and  $g_R$  depend on  $s, t, u$  and can be precomputed, but for better understanding we update the values on-line and store them in  $T$ . We first describe how to handle the top front while traversing the grid. Initially,  $F_T = G(s, t)$ . As long as we have more top sites than top ports in  $K$ , we can connect all ports to sites and thus can maintain  $F_T = G(s, t)$ . As soon as we have exactly the same amount of top ports and top sites in  $K$  and we encounter a port event for a top port, we have to check the strip condition and find the rightmost point  $F_T$  with  $y(F_T) = G_y(t)$  such that the rectangle  $R_{F_T}$  spanned by  $F_T$  and  $r$  is valid. By storing  $F_T$ , we know that all ports to the right of  $x(F_T)$  are already connected to a site, all sites to the top-right of  $F_T$  are already connected to a port, and all top sites to the bottom-left of  $F_T$  have to be connected to a port that lies left of  $x(F_T)$ . Thus we do not have to check new strip conditions until  $s < x(F_T)$ . We handle  $F_R$  similarly.

We now look at the length of the top leaders, the length of the right leaders can be handled similarly. Note that by moving from  $t$  to  $t - 1$ , the length of the top leaders does not change. If  $g_T > 0$ , we imagine an additional port at  $x(F_T)$  that can be connected to  $g_T$  top sites. When moving from  $s$  to  $s - 1$ , we add  $g_T \cdot (G_x(s) - G_x(s - 1))$  to  $l$ . When we calculate a new value  $F_T$  by checking the strip condition, we can already connect all top sites inside the top front to top ports, and add the corresponding leader length to  $l$ . Thus, we can only encounter site events for sites that are a) inside  $F_T \setminus K$  or b) have to be connected to a top port that lies left of  $x(F_T)$ . In case a) we do not change  $l$ , in case b) we connect the site to the imaginary port, add the length of the corresponding leader to  $l$  and increase  $g_T$  by 1. When we encounter a port event, if the port lies inside  $F_T \setminus K$ , we do not change  $l$ , otherwise we can connect any of the unlabeled sites to this port. We add the horizontal distance between  $G_x(s)$  and the port to  $l$  and decrease  $g_T$  by 1. Note that by choosing any unlabeled site, the resulting solution may not be planar. However, because the bends of all unconnected sites will be above  $C$ , we can use Lemma 1 to remove

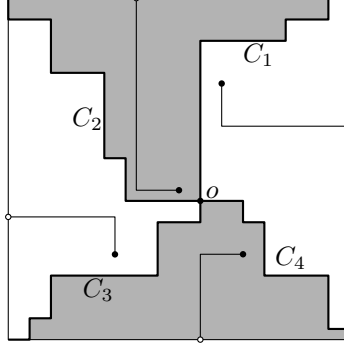


Fig. 12: The curves  $C_1$ ,  $C_2$ ,  $C_3$  and  $C_4$  meeting at the point  $o$  partition the rectangle into four regions.

the crossings without changing the total leader length.

Since the matrix now has four additional fields, the running time and storage is increased by a factor of four. Additionally, we need  $O(n \log n)$  time to check the strip condition and get a length-minimal solution for the sites and ports inside  $F_T \setminus K$  and  $F_R \setminus K$ . Note that, by using an appropriate data structure to precompute the fronts, the running time can be slightly decreased.

**Theorem 5.** TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES can be solved in  $O(n^8 \log n)$  time using  $O(n^6)$  space such that the total leader length is minimized.

## 5 Three- and Four-sided Boundary Labeling

In this section, we also allow labels on the bottom side and the left side of  $R$ . In order to solve an instance of the three- and four-sided case, we adapt the techniques we developed for the two-sided case. We start with the four-sided case as the most general case and we then show how the presented approach can be used to solve the three-sided case. We assume that the ports are fixed.

### 5.1 Four-Sided Planar Solution

Similar to the two-sided boundary labeling, we pursue the idea that if there exists a planar solution, then we also can find a planar solution such that there are four  $xy$ -monotone curves going from the four corners of  $R$  to a common point  $o$ , such that these curves separate the leaders of the different label types from each other; see Fig. 12. To that end, we first show that leaders of left and right labels can be separated vertically and leaders of top and bottom labels can be separated horizontally. Afterwards, we apply the result of Lemma 2 in order to resolve the remaining overlaps, e.g., between top and right leaders. Then, based on these results, we present a simple algorithm that systematically explores different partitions of the solutions that can be solved by the algorithm for the TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES.

**Definition 2.** A planar solution for the four-sided boundary labeling problem is  $x$ -separated if there exists a vertical line  $\ell$  such that

- a) the sites that are labeled to the left side are to the left of  $\ell$ ,
- b) the sites that are labeled to the right side are to the right of  $\ell$ .

A planar solution is  $y$ -separated if there exists a horizontal line  $\ell$  such that

- a) the sites that are labeled to the top side are above  $\ell$
- b) the sites that are labeled to the bottom side are below  $\ell$ .

The following lemma proves that we can always find a planar solution that is both  $x$ -separated and  $y$ -separated, if a solution exists.

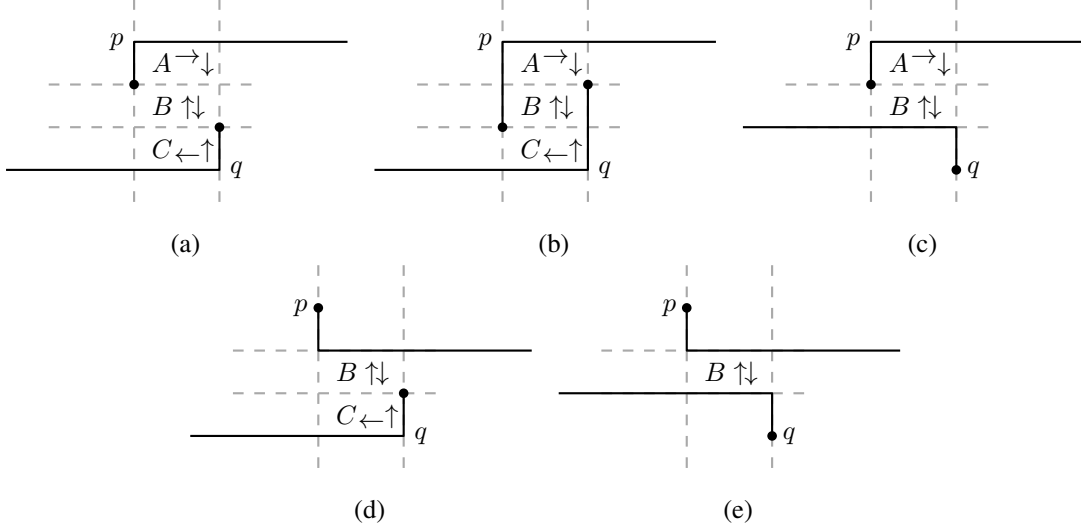


Fig. 13: All possible horizontal overlaps of a right and left leader when assuming that the right leader lies above the left leader.

**Lemma 6.** *If there exists a planar solution  $\mathcal{L}$  for the four-sided boundary labeling problem, then there exists a planar solution  $\mathcal{L}'$  that is both  $x$ -separated and  $y$ -separated, such that  $|\mathcal{L}'|_x \leq |\mathcal{L}|_x$  and  $|\mathcal{L}'|_y \leq |\mathcal{L}|_y$ .*

*Proof.* We show that if  $\mathcal{L}$  is not  $x$ -separated, then we can decrease either  $|\mathcal{L}|_x$  or  $|\mathcal{L}|_y$  without increasing the other one. A symmetric argument holds for instances that are not  $y$ -separated.

Assume that  $\mathcal{L}$  is not  $x$ -separated. Then there exist sites  $p$  and  $q$  with  $x(q) > x(p)$ , such that  $p$  is labeled by a right port  $r$ , and  $q$  is labeled by a left port  $\ell$ . We choose  $p$  and  $q$  as a minimal pair in the sense that their leaders have minimum vertical distance among all such pairs. Without loss of generality, assume that  $\lambda(p, r)$  is above  $\lambda(q, \ell)$ , otherwise we mirror the instance vertically. We now describe possible reroutings to reduce  $|\mathcal{L}|_x$ .

Let  $S$  denote the vertical strip bounded by vertical lines through  $p$  and  $q$ . Now consider the horizontal lines through  $\text{bend}(p, r)$ ,  $p$ ,  $q$ , and  $\text{bend}(q, \ell)$ . These lines cut out three bounded rectangles from  $S$ ; see Fig. 13. If the topmost of these rectangles is below  $\lambda(p, r)$ , we denote it by  $A$ , otherwise we set  $A = \emptyset$ . Symmetrically, if the bottommost of these rectangles is above  $\lambda(q, \ell)$ , we denote it by  $C$ , otherwise we set  $C = \emptyset$ . The middle rectangle is denoted by  $B$ . Due to the choice of  $p$  and  $q$  with minimum vertical distance, and the planarity of  $\mathcal{L}$ , these rectangles satisfy the following properties.

- (i) All leaders that intersect  $A$  are either right or bottom leaders.
- (ii) All leaders that intersect  $B$  are either top or bottom leaders.
- (iii) All leaders that intersect  $C$  are either left or top leaders.

We now distinguish cases based on which types of leaders intersect these rectangles. We show that in each case we can find a new planar solution that decreases at least one of  $|\mathcal{L}|_x$  and  $|\mathcal{L}|_y$  without increasing the other one.

**Case 1:** All leaders intersecting  $A$  are connected to the right, all leaders intersecting  $C$  are connected to the left, and  $B$  is not intersected by any leaders; see Fig. 14c. We reroute  $p$  to  $\ell$  and  $q$  to  $r$ , which decreases  $|\mathcal{L}|_x$  without increasing  $|\mathcal{L}|_y$ . This may create crossings, but they all lie either on the right side of  $A$  or on the left side of  $C$ . After the rerouting, all leaders (including the new ones) that intersect  $A$  are right leaders and have their bend in  $A$ . A symmetric statement holds for  $C$ . We can thus apply Lemma 1 twice to obtain a planar solution without increasing  $|\mathcal{L}|_x$  or  $|\mathcal{L}|_y$ .

**Case 2:** All leaders intersecting  $A$  are to the right, all leaders intersecting  $C$  are to the left, but  $B$  is intersected by some leaders; see Fig. 14a. Note that this case can only occur if  $p$  lies above  $q$ , as otherwise the sites labeled by these leaders would contradict the minimality of  $p$  and  $q$ .

In a first step, we “sort” the leaders intersecting  $B$  vertically as follows. Let  $\lambda_t$  and  $\lambda_b$  be the lowest

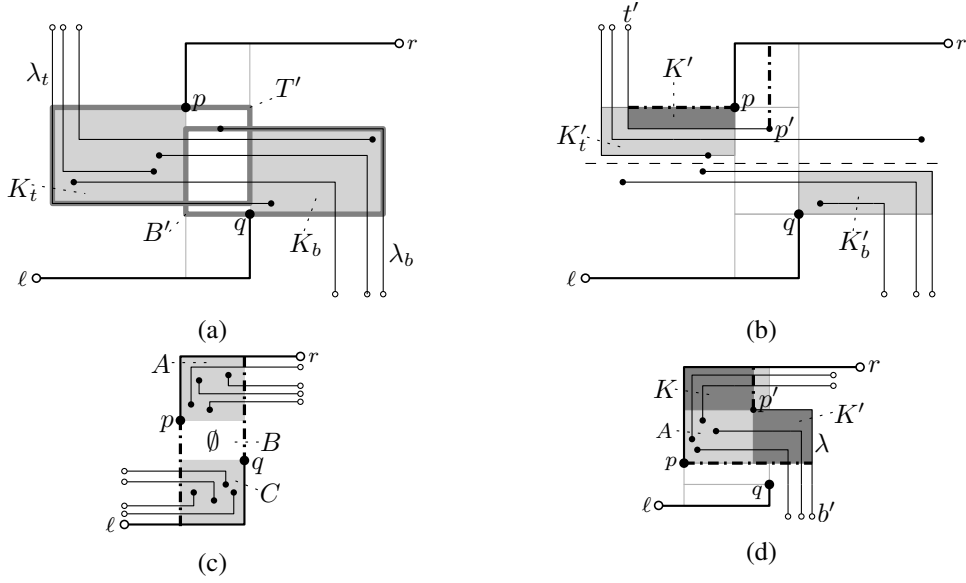


Fig. 14: Different constellations of leaders intersecting the rectangles  $A$ ,  $B$  and  $C$ ; see Fig. 13. a)  $B$  is intersected by top and bottom leaders. b)  $B$  is intersected by top and bottom leaders such that there is a horizontal line separating both types. c)  $B$  is empty, while  $A$  is only intersected by right leaders and  $C$  is only intersected by left leaders. d)  $A$  is intersected by right and bottom leaders.

top and highest bottom leader that intersects  $B$ , respectively. Let  $T'$  be the set of top leaders that intersect the rectangle spanned by  $\text{bend}(\lambda_t)$  and the top right corner of  $B$ . Symmetrically,  $B'$  is the set of bottom leaders that intersect the rectangle spanned by  $\text{bend}(\lambda_b)$  and the bottom left corner of  $B$ . Further, let  $K_t$  be the rectangle spanned by  $\text{bend}(\lambda_t)$  and  $p$ , and let  $K_b$  be the rectangle spanned by  $\text{bend}(\lambda_b)$  and  $q$ . Note that the only leaders that intersect  $K_t$  or  $K_b$  are top and bottom leaders; left or right leaders would contradict the minimality of  $p$  and  $q$ . Moreover, any such leader is contained in either  $T'$  or  $B'$ .

Denote by  $P'$  the sites labeled by the leaders in  $T' \cup B'$ . We reroute the leaders in  $T' \cup B'$  arbitrarily, such that the  $|T'|$  topmost sites of  $P'$  are labeled by top labels, and the remaining  $|B'|$  points on the bottom are connected to the bottom side. Denote the resulting set of leaders by  $T''$  (top leaders) and  $B''$  (bottom leaders). Note that this rewiring does not increase  $|\mathcal{L}|_x$  and  $|\mathcal{L}|_y$ , and it may in fact decrease  $|\mathcal{L}|_y$ . Next, we deal with any crossings that may have resulted from the reordering. Note that we have only reordered leaders with bends in either  $K_t$  or  $K_b$ , thus new intersection may only arise in the interior of these rectangles (by Claim 1 of Lemma 1).

Consider the rectangle  $K'_t$  spanned by the bend of the lowest leader in  $T''$  and  $p$ . Due to the rerouting, we have that the lower left corner of  $K'_t$  is contained in  $K_t$ , and thus we have  $K'_t \subseteq K_t$ . Thus, leaders that intersect  $K'_t$  are either top or bottom leaders. However, the lowest top leader in  $T''$  lies above the highest bottom leader in  $B''$ , which implies that no bottom leader intersects  $K'_t$ . We can thus apply Lemma 1 to remove any intersection in  $K'_t$ . A symmetric argument holds for  $K'_b$ . This yields a planar solution without increasing  $|\mathcal{L}|_x$  or  $|\mathcal{L}|_y$ . Figure 14b illustrates the structure of such a solution  $\mathcal{L}''$ .

If in  $\mathcal{L}''$  no leader intersects  $B$ , we proceed as in Case 1. Hence, assume that there is a leader intersecting  $B$ , and without loss of generality, it is a top leader, otherwise we rotate the instance by  $180^\circ$ . Note that such a leader labels a site that is right of  $p$ . Let  $p'$  be the topmost such site that is connected to the top and lies right of  $p$ . Let  $t'$  denote the port of the label of  $p'$ . We reroute  $r$  to  $p'$  and  $t'$  to  $p$ . This strictly decreases both  $|\mathcal{L}''|_x$  and  $|\mathcal{L}''|_y$ . Crossings may arise only on the vertical segment of  $\lambda(p', r)$  and the horizontal segment of  $\lambda(p, t')$ . Crossings on  $\lambda(p', r)$  may only occur with leaders that intersect  $A$ . Then, by assumption of case 2, all leaders involved in these crossings are right leaders having their bends inside  $A$ . We use Lemma 1 to remove the crossings.

The crossings on  $\lambda(p, t')$  lie inside the rectangle  $K'$  spanned by  $\text{bend}(\lambda(p', t'))$  and  $p$ . A bottom leader may not cross  $K'$  due to the ordering of the leaders intersecting  $B$ . A left leader intersecting  $K'$

would have intersected  $\lambda(p', t')$  before the rerouting, and finally, a right leader intersecting  $K'$  would contradict the minimality of  $p$  and  $q$ . Therefore, all leaders intersecting  $K'$  are top leaders. Moreover, all leaders intersecting  $K'$  have their bend in  $K'$ , otherwise they would have intersected  $\lambda(p', t')$  before the rerouting. Again, we apply Lemma 1 to remove all crossings, and obtain a planar solution. Overall, we have reduced  $|\mathcal{L}|_x$  and  $|\mathcal{L}|_y$ .

**Case 3:** The rectangle  $A$  is intersected by a bottom leader or  $C$  is intersected by a top leader. Without loss of generality, we assume that  $A$  is intersected by a bottom leader, otherwise we rotate the instance by  $180^\circ$ ; see Fig. 14d.

Let  $\lambda$  denote the topmost bottom leader intersecting  $A$ . Then  $\lambda$  labels a site  $p'$  inside  $A$ . Let  $b'$  be the corresponding port on the bottom side. We reroute  $p'$  to  $r$  and  $p$  to  $b'$ . Observe that this decreases  $|\mathcal{L}|_x$  and  $|\mathcal{L}|_y$ , but may introduce crossings. All crossings lie on either the vertical segment of  $\lambda(p', r)$  or the horizontal segment of  $\lambda(p, b')$ . We again use Lemma 1 to remove the crossings.

Let  $K$  denote the rectangle spanned by  $\text{bend}(p, r)$  and  $p'$ . By the choice of  $p'$ , all leaders intersecting  $K$  are right leaders. As they do not cross  $\lambda(p, r)$ , their bends are contained in  $K$  as well. Lemma 1 can be applied. For the crossings on  $\lambda(p, b')$  consider the rectangle  $K'$  spanned by  $p'$  and  $\text{bend}(p, b')$ . The top side is open, such that  $K'$  does not contain  $p'$ . All leaders that intersect  $K'$  are bottom leaders. Top leaders would have crossed  $\lambda(p, r)$ , left leaders would contradict the minimality of  $p$  and  $q$ , and right leaders would have crossed  $\lambda(p', b')$ . Moreover, all these leaders have their bends inside  $K'$  by the choice of  $p'$  as the highest site connected to the bottom. Again Lemma 1 applied to  $K'$  removes the remaining crossings, and in total we have reduced  $|\mathcal{L}|_x$  and  $|\mathcal{L}|_y$ .  $\square$

This lemma shows that when searching for a planar solution of the labeling problem, we can restrict ourselves to solutions that are  $x$ -separated and  $y$ -separated. Let  $\mathcal{L}$  denote such a solution, and let  $\ell_v$  and  $\ell_h$  be the lines separating the sites labeled by left and right labels, and the ones labeled by top and bottom labels, respectively. Let  $o \in R$  denote the intersection of  $\ell_v$  and  $\ell_h$ , called *center point*. Let  $r_1, \dots, r_4$  denote the corners of  $R$ , named in counterclockwise ordering, and such that  $r_1$  is the top right corner. Consider the rectangles that are spanned by  $o$  and  $r_i$  for  $i = 1, \dots, 4$ . Each of them contains only two types of leaders. For example the top right rectangle contains only top and right leaders. An  $x$ - and  $y$ -separated planar solution is *partitioned* if, for each rectangle spanned by  $o$  and one of the corners  $r_i$  of  $R$ , there exists an  $xy$ -monotone curve  $C_i$  from  $r_i$  to  $o$  that separates the two different types of leaders contained in that rectangle; see Fig. 12. Our next step is to show that a planar solution can be transformed into a partitioned solution without increasing  $|\mathcal{L}|_x$  and  $|\mathcal{L}|_y$ .

**Proposition 2.** *If there exists a planar solution  $\mathcal{L}$  for FOUR-SIDED BOUNDARY LABELING, then there is also a partitioned solution  $\mathcal{L}'$ .*

*Proof.* By Lemma 6, we can assume that  $\mathcal{L}$  is  $x$ - and  $y$ -separated. Let  $o$  be the center point as defined above. We show how to ensure that the rectangle  $K$  spanned by  $o$  and  $r_1$  admits a curve that separates the leaders inside it. The remaining cases are symmetric.

Essentially, we proceed as in the proof of Lemma 2 to remove obstructions of types (P1)–(P4); see Fig. 5. We note that in the rerouting, we only exchange a top label with a right label, and hence the solution remains  $x$ - and  $y$ -separated. Moreover, leader changes are inside  $K$ , and thus we can remove all patterns in  $K$  without interfering with the remaining quadrants. After all patterns have been removed, a curve connecting the top right corner of  $R$  to  $c$ , separating the top labels from the right labels can be found as in the proof of Lemma 2.  $\square$

To compute such a partitioned solution  $\mathcal{L}$ , assume we are given, for each side  $s \in \{L(\text{left}), R(\text{right}), T(\text{top}), B(\text{ottom})\}$  of the rectangle  $R$ , the leader  $\lambda_s$  of  $\mathcal{L}$  whose segment orthogonal to  $s$  is maximum among all leaders of side  $s$ . These *extremal leaders* essentially partition the instance into four smaller instances of ADJACENT TWO-SIDED BOUNDARY LABELING, one for each corner; see Fig. 15. The idea is then to solve these instances independently using an adaption of our dynamic program as described in Section 3. In order to obtain the extremal leaders  $\lambda_s$  we just explore all  $O(n^8)$  possible



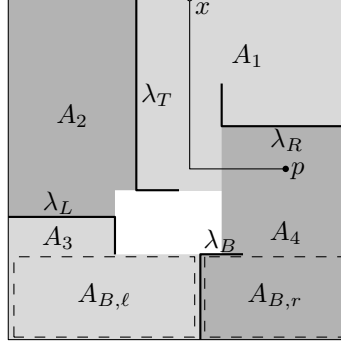


Fig. 15: The extremal leaders  $\lambda_L$ ,  $\lambda_R$ ,  $\lambda_T$  and  $\lambda_B$  partition the rectangle  $R$  into the four areas  $A_1$ ,  $A_2$ ,  $A_3$  and  $A_4$ , which can be considered independently. The leader  $\lambda(p, x)$  illustrates Lemma 8.

tuples  $(\lambda_L, \lambda_R, \lambda_T, \lambda_B)$  of leaders. Trying all of them thus yields a running time of  $O(n^{10})$  and space consumption  $O(n)$ .

We now describe the approach in detail. To that end assume, that we are given a partitioned solution  $\mathcal{L}$  and let  $\lambda_L$ ,  $\lambda_R$ ,  $\lambda_T$ ,  $\lambda_B$  be the extremal leaders of the four sides of  $R$ , respectively. In order to define the four instances induced by those leaders formally, let  $A_{s,\ell}$  be the rectangle that is spanned by  $\text{bend}(\lambda_s)$  and the corner of  $R$  on  $s$  that lies to the left of  $\lambda_s$  when looking from the port of  $\lambda_s$  to  $\text{bend}(\lambda_s)$ ; see Fig. 15. Analogously, let  $A_{s,r}$  be the rectangle that is spanned by  $\text{bend}(\lambda_s)$  and the corner of  $R$  on  $s$  that lies to the right of  $\lambda_s$  when looking from the port of  $\lambda_s$  to  $\text{bend}(\lambda_s)$ . We further assume that  $\lambda_s$  neither belongs to  $A_{s,\ell}$  nor to  $A_{s,r}$ .

Then the instance  $I_1$ , which is located at the upper right corner of  $R$ , contains all ports and sites that lie in  $A_1 = (A_{T,\ell} \cup A_{R,r}) \setminus (A_{T,r} \cup A_{R,\ell})$ ; see Fig. 15. For the remaining corners of  $R$  we similarly define in counterclockwise order the instances  $I_2$ ,  $I_3$  and  $I_4$  using the sets  $A_2 = (A_{T,r} \cup A_{L,\ell}) \setminus (A_{T,\ell} \cup A_{L,r})$ ,  $A_3 = (A_{L,r} \cup A_{B,\ell}) \setminus (A_{L,\ell} \cup A_{B,r})$  and  $A_4 = (A_{B,r} \cup A_{R,\ell}) \setminus (A_{B,\ell} \cup A_{R,r})$ , respectively. We first show that these areas do not overlap in a partitioned solution.

**Lemma 7.** *For a partitioned solution  $\mathcal{L}$  the sets  $A_1$ ,  $A_2$ ,  $A_3$  and  $A_4$  are pairwise disjoint.*

*Proof.* Since the solution  $\mathcal{L}$  is partitioned, by definition there is a vertical line  $\ell_v$  that separates  $\lambda_R$  from  $\lambda_L$  and there is a horizontal line  $\ell_h$  that separates  $\lambda_T$  from  $\lambda_B$ . For the sake of contradiction assume that there are two sets  $A_i$  and  $A_j$  with  $i \neq j$  that overlap.

First assume that  $A_i$  and  $A_j$  belong to adjacent corners. Without loss of generality we consider  $A_1 = (A_{T,\ell} \cup A_{R,r}) \setminus (A_{T,r} \cup A_{R,\ell})$  and  $A_2 = (A_{T,r} \cup A_{L,\ell}) \setminus (A_{T,\ell} \cup A_{L,r})$ . Symmetric arguments hold for the other cases. By definition,  $A_{T,\ell}$  and  $A_{T,r}$  are disjoint. Thus, the sets  $A_{R,r}$  and  $A_{L,\ell}$  must overlap. As, by definition,  $A_{R,r}$  lies to the right of the vertical line through the site of  $\lambda_R$  and  $A_{L,\ell}$  lies to the left of the vertical line through the site of  $\lambda_L$ , the vertical line  $\ell_v$  also separates  $A_{R,\ell}$  and  $A_{L,\ell}$ .

Now assume that  $A_i$  and  $A_j$  belong to corners that are not adjacent. Without loss of generality, we consider  $A_1 = (A_{T,\ell} \cup A_{R,r}) \setminus (A_{T,r} \cup A_{R,\ell})$  and  $A_3 = (A_{L,r} \cup A_{B,\ell}) \setminus (A_{L,\ell} \cup A_{B,r})$ . Again, by definition,  $A_{T,\ell}$  lies above  $\ell_h$ ,  $A_{B,\ell}$  lies below  $\ell_h$ ,  $A_{R,r}$  lies to the right of  $\ell_v$  and  $A_{L,r}$  lies to the left of  $\ell_v$ . Thus,  $A_{T,\ell}$  and  $A_{B,\ell}$  are disjoint as well as  $A_{R,r}$  and  $A_{L,r}$ . The sets  $A_{B,\ell}$  and  $A_{R,r}$  are disjoint because due to the previous case the sets  $A_{B,r}$  and  $A_{R,r}$  are disjoint. A symmetric argument holds for  $A_{T,\ell}$  and  $A_{R,r}$ .  $\square$

The next lemma shows that if we split a partitioned solution  $\mathcal{L}$  into the four sub-instances as defined above, then the sites contained in  $I_1$  are connected to ports in  $I_1$ . Symmetric statements can be derived for  $I_2$ ,  $I_3$  and  $I_4$ . In particular, this shows that we can find  $\mathcal{L}$  only considering the sub-instances.

**Lemma 8.** *For a partitioned solution  $\mathcal{L}$  all sites connected to ports of instance  $I_1$  are contained in  $A_1$ .*

*Proof.* Due to the extremal choice of  $\lambda_T$  and  $\lambda_R$ , all sites connected to ports of  $I_1$  must be contained in  $A_{T,\ell} \cup A_{R,r}$ . Assume that there is a port  $x$  in  $I_1$  that is connected to a site  $p$  that lies in  $(A_{T,\ell} \cup A_{R,r}) \cap (A_{T,r} \cup A_{R,\ell})$ . Without loss of generality, let  $x$  be a top port. Symmetric arguments hold for right ports.

As  $p$  is connected to a top port, it cannot be contained in  $A_{T,r}$ , because otherwise it would intersect  $\lambda_T$ . Thus, it must lie in  $A_{R,\ell}$ ; see Fig. 15. Note that this also implies that  $\lambda_R$  lies completely in  $A_{T,\ell}$ , because otherwise  $(A_{T,\ell} \cup A_{R,r}) \cap A_{R,\ell} = \emptyset$ . As therefore  $p$  lies below and to the right of the site of  $\lambda_R$  and  $x$  lies above and to the left of the site of  $\lambda_R$ , the leaders  $\lambda_R$  and  $\lambda(p, x)$  must form an overlap of type (P1) or (P2); see Fig. 5. This is a contradiction to Lemma 2 and the assumption that  $\mathcal{L}$  is partitioned.  $\square$

Now we can apply the following approach: We explore all  $O(n^8)$  possible tuples  $(\lambda_L, \lambda_R, \lambda_T, \lambda_B)$  of leaders. For each tuple, we check whether it permits a partitioned solution, i.e., whether these four leaders are  $x$ - and  $y$ -separated. This obviously needs  $O(1)$  time. In case of a negative result, we continue with the next tuple of leaders. Otherwise, due to Lemma 7 and Lemma 8 it is sufficient to independently consider the four instances  $I_1, I_2, I_3$  and  $I_4$  that are induced by  $(\lambda_L, \lambda_R, \lambda_T, \lambda_B)$  as described above. If all four instances could be solved, there exists a solution, otherwise not.

We now describe how to adapt the dynamic program of Section 3 for the instance  $I_1$ , i.e., in the following we only consider ports and sites that belong to  $I_1$ . The other instances can be solved symmetrically. Let  $G$  be the underlying grid as defined in Section 3 restricted to ports and sites in  $I_1$  and the ports and sites of  $\lambda_T$  and  $\lambda_R$ , and let  $\mathbf{c} := (s, t)$  be the current grid point we checked as endpoint for  $C$ .

Based on Section 3, we extend the description for obtaining the table entry  $T[\mathbf{c} - \Delta\mathbf{c}, \cdot]$  for  $\Delta\mathbf{c} = (1, 0)$ , i.e, we decrease  $s$  by 1. The other case  $\Delta\mathbf{c} = (0, 1)$  is completely symmetric. Assume  $T[\vec{c}, u] = \text{true}$ . First, we apply the following check. If  $G(s-1, t)$  lies in the rectangle spanned by the site of  $\lambda_T$  and the upper left corner of  $R$ , we set  $T[\mathbf{c} - \Delta\mathbf{c}, u] = \text{false}$  for all  $1 \leq u \leq n$ , because in that case  $C$  cannot reach the grid point  $(0, 0)$  without intersecting  $\lambda_T$ . Otherwise, we deal with site and port events as follows.

First, assume going from  $s$  to  $s-1$  is a site event, i.e., there is a site  $p$  with  $G_x(s) > x(p) > G_x(s-1)$ . We first check on the position of  $p$ . If  $p$  is located above  $C$  and does not lie in  $A_{T,\ell}$ , we set  $T[\mathbf{c} - \Delta\mathbf{c}, u] = \text{false}$  for all  $1 \leq u \leq n$ , because due to the extremal choice of  $\lambda_T$  the site  $p$  cannot be connected to any top port. Analogously, if  $p$  is located below  $C$  and does not lie in  $A_{R,r}$ , we set  $T[\mathbf{c} - \Delta\mathbf{c}, u] = \text{false}$  for all  $1 \leq u \leq n$ , because due to the extremal choice of  $\lambda_R$  the site  $p$  cannot be connected to any right port. If none of both cases occur, we proceed as originally described for site-events, otherwise we compute the next table entry. The treatment for port events is left unchanged.

Note that, by definition of the dynamic program, the curve  $C$  can neither intersect  $\lambda_T$  nor  $\lambda_R$ , but it separates both leaders. Thus, in any solution  $\mathcal{L}$  obtained by the dynamic program the leaders of the right ports cannot intersect  $\lambda_T$  and the leaders of the top ports cannot intersect  $\lambda_R$ . Further, by the extremal choice of  $\lambda_T$  and  $\lambda_R$ , the leaders of the right ports cannot intersect  $\lambda_R$  and the leaders of the top ports cannot intersect  $\lambda_T$ . All together, we obtain a crossing free solution for  $I_1$ , if it exists.

**Theorem 6.** FOUR-SIDED BOUNDARY LABELING can be solved in  $O(n^{10})$  time using  $O(n)$  space.

Note that, except for the length minimization, all presented extensions in Section 4 carry over, as we only solve sub-instances of the TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES. The running times increase correspondingly.

## 5.2 Three-Sided Planar Solution

For THREE-SIDED BOUNDARY LABELING, we can apply the same approach as for FOUR-SIDED BOUNDARY LABELING, obtaining a running time  $O(n^8)$  when only considering three extremal leaders. Consequently, all extensions for FOUR-SIDED BOUNDARY LABELING carry over.

In this section, we show how this result can be improved to  $O(n^4)$  time by guessing only the extremal leader of the middle side of the rectangle  $R$ . Note that we can imagine an instance of THREE-SIDED BOUNDARY LABELING as a degenerated instance of the FOUR-SIDED BOUNDARY LABELING where bottom ports are missing. Proposition 2 therefore holds also for the three-sided case when assuming that the four curves partitioning the solution meet on the bottom segment of  $R$ . Thus, for a partitioned solution  $\mathcal{L}$  we can restrict ourselves to the curves  $C_1$  and  $C_2$  as defined in the section before; see Fig. 16.

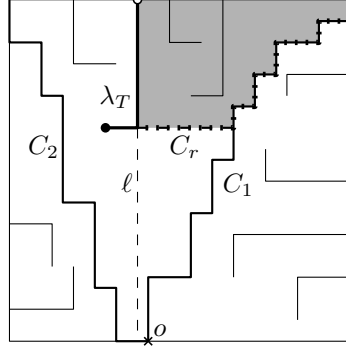


Fig. 16: The two curves  $C_1$  and  $C_2$  meeting in point  $o$  partitioning the solution. The extremal top leader  $\lambda_T$  induces a separation of the leaders in the gray area and the remaining leaders. The resulting curve  $C_r$  is dotted. The line  $\ell$  separates the instance  $I_r$  from  $I_\ell$ .

Further, we can describe the structure of  $\mathcal{L}$  by means of the extremal top leader  $\lambda_T$  of  $\mathcal{L}$ . We call an  $xy$ -monotone curve  $C$  connecting the top-right corner of  $R$  with  $\text{bend}(\lambda_T)$  an  $\lambda_T$ -separating curve with respect to  $\mathcal{L}$  if the area  $A$  above  $C$  enclosed by  $C$ ,  $\lambda_T$  and the top side of  $R$  contains only top leaders of  $\mathcal{L}$  with ports in  $A$ .

**Lemma 9.** *Let  $\mathcal{L}$  be a partitioned solution. Then there is an  $\lambda_T$ -separating curve  $C_r$  with respect to  $\mathcal{L}$  connecting the top-right corner of  $R$  with  $\text{bend}(\lambda_T)$ .*

*Proof.* Let  $C_1$  be the curve of the partition of  $\mathcal{L}$  that goes from the top-right corner of  $R$  to the bottom side of  $R$ . We construct  $C_r$  based on  $C_1$ . To that end, let  $\ell$  be the horizontal half-line emanating from  $\text{bend}(\lambda_T)$  to the right. We denote the first intersection point of  $\ell$  and  $C_1$  by  $p$ . We then construct the curve  $C_r$  by the horizontal line segment  $\overline{p \text{bend}(\lambda_T)}$  and the part of  $C_1$  connecting  $p$  with the top-right corner of  $R$ ; see Fig. 16. As  $\lambda_T$  is extremal under all top leaders of  $\mathcal{L}$ , we can directly conclude that the enclosed area  $A$  as defined above contains only top leaders with ports in  $A$ . In particular, as all right leaders lie below  $C_1$ , they also must lie below  $C_r$ .  $\square$

Analogously to the four-sided case, we explore all  $O(n^2)$  choices of leaders in order to find an appropriate extremal top leader. Let  $\lambda_T$  be the current choice. Then we apply the following approach.

We denote the rectangle of  $R$  spanned by the bottom-right corner of  $R$  and the port of  $\lambda_T$  by  $S_r$ . Removing the site and port of  $\lambda_T$  from  $S_r$ , the instance  $I_r$  is defined by the ports and sites contained in  $S_r$ . Symmetrically, we can define the strip  $S_\ell$  and the instance  $I_\ell$  based on the bottom-left corner of  $R$  and the port of  $\lambda_T$ ; see Fig. 16.

Obviously,  $I_r$  and  $I_\ell$  are two independent instances of TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES. We now apply the dynamic program of Section 3 to solve  $I_r$ . To that end, we only compute the table entries that are contained in the rectangle spanned by the top-right corner of  $R$  and  $\text{bend}(\lambda_T)$ ; see Fig. 17a. Without loss of generality, we assume that  $\text{bend}(\lambda_T)$  coincides with the origin of the underlying grid  $G$ . We then are interested in the solution for the entry  $T[0, 0, n_r]$ , where  $n_r$  denotes the number of top ports in  $I_r$ . If  $T[0, 0, n_r] = \text{false}$ , we know by Lemma 9 that there is no solution and we can continue with the next choice of  $\lambda_T$ . Otherwise, by means of  $T[0, 0, n_r]$ , we obtain an  $\lambda_T$ -separating curve  $C_r$  connecting the top-right corner of  $R$  with  $\text{bend}(\lambda_T)$  and an  $xy$ -separated solution  $\mathcal{L}_r$  with respect to  $C_r$ . This solution comprises all ports and sites of  $I_r$  that have been considered by the dynamic program, i.e., each port and site of those is labeled. Note that in general there are ports and sites in  $I_r$  that are not labeled with respect to  $\mathcal{L}_r$ . We call those ports and sites *free ports* and *free sites* with respect to a certain solution, namely in that case with respect to  $\mathcal{L}_r$ ; see Fig. 17. Let  $B_r$  be the bounding box of the right leaders in  $\mathcal{L}_r$ , i.e., the smallest axis-aligned rectangle that contains all right leaders of  $\mathcal{L}_r$ , then the next lemma describes where free ports and labels may occur.

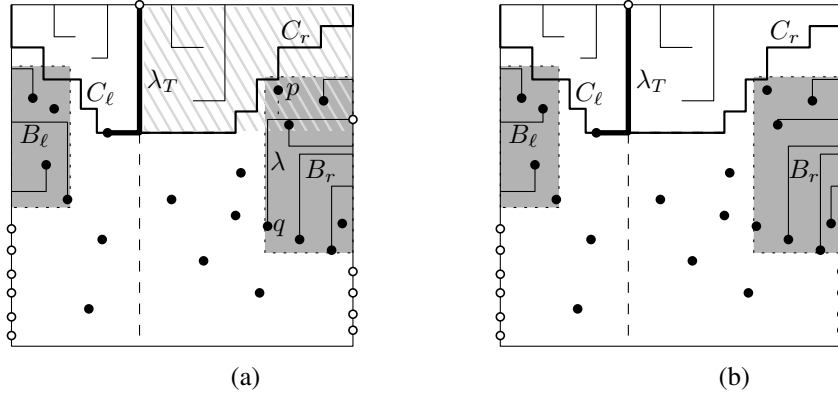


Fig. 17: a) An instance after applying the dynamic program. The light gray shaded rectangle spanned by the top-right corner of  $R$  and  $\text{bend}(\lambda_T)$  comprises the table entries which are computed. The dashed line separates the instance  $I_\ell$  and  $I_r$ . In both instances, there are free sites and ports. The bounding boxes  $B_\ell$  and  $B_r$  are depicted in gray. b) The same instance after applying `MakeReachable`.

**Lemma 10.** *All free sites of  $I_r$  with respect to  $\mathcal{L}_r$  lie below  $C_r$  and all free ports of  $I_r$  with respect to  $\mathcal{L}_r$  lie below  $B_r$  and below the horizontal line through  $\text{bend}(\lambda_T)$ . In particular, the free ports are right ports.*

*Proof.* Due to  $T[0, 0, n_r] = \text{true}$  and the choice of  $\mathcal{L}_r$ , all  $n_r$  sites above  $C_r$  must be connected to the  $n_r$  top ports of  $I_r$ . Thus, only sites below  $C_r$  and only right ports are free. It remains to show that all free ports lie below  $B_r$  and the horizontal line through  $\text{bend}(\lambda_T)$ . First, note that all right ports above the horizontal line  $\ell$  through  $\text{bend}(\lambda_T)$  are connected to a site in  $\mathcal{L}_r$ , because they have been used as port events when applying the dynamic program.

Assume for a contradiction that there is a free port  $x$  that lies above the bottom segment of  $B_r$ . Then there is a leader  $\lambda$  in  $\mathcal{L}_r$  that reaches a site below  $x$ . As due to the previous reasoning  $x$  must lie below  $\ell$ , this leader must be a right leader. Thus, the port  $x$  lies within the rectangle spanned by the port and site of  $\lambda$ . But in order to satisfy the strip condition, the port must have been considered by the dynamic program, which is a contradiction to the fact that  $x$  is free.  $\square$

Due to symmetry, we can apply the same approach on  $I_\ell$  in order to obtain a solution  $\mathcal{L}_\ell$  and an  $\lambda_T$ -separating curve  $C_\ell$  connecting the top-left corner of  $R$  with  $\text{bend}(\lambda_T)$ . An analogous statement as in the lemma above holds for  $\mathcal{L}_\ell$ .

We now describe how the remaining free ports and sites can be connected such that no crossings arise and the curves  $C_r$  and  $C_\ell$  remain  $\lambda_T$ -separating. To that end, we make the following assumption:

A1:  $I_r$  contains more free sites than free ports with respect to  $\mathcal{L}_r$ .

The opposite case can be solved by just reflecting the whole instance, because this assumption also means that  $\mathcal{L}_\ell$  has more free ports than free sites with respect to  $\mathcal{L}_\ell$ . Further, A1 implies that we need to connect left ports with sites of  $I_r$ . For an example see Fig. 17a.

We first ensure that all free sites of  $I_r$  with respect to  $\mathcal{L}_r$  are *reachable* from all free ports of  $I_r$  with respect to  $\mathcal{L}_r$ , i.e., we ensure that if connecting a free port with a free site, we do not create any crossing with leaders of  $\mathcal{L}_r$ .

As due to Lemma 10 all free ports of  $\mathcal{L}_r$  lie below  $B_r$ , all sites below  $B_r$  are reachable from free ports of  $I_r$ . For the remaining free sites in  $B_r$ , we apply the following algorithm, which we call `MakeReachable`.

We add all free sites in  $B_r$  to a queue  $Q$  sorted by their  $x$ -coordinate from right to left. As long as  $Q$  is not empty, we remove the next rightmost site  $p$  from  $Q$  and emanate a vertical half-line  $\ell_d$  from  $p$  downwards. The first time  $\ell_d$  hits a right leader  $\lambda$ , we apply a rewiring, namely we connect the port

of  $\lambda$  with  $p$  obtaining the new leader  $\lambda'$ . Note that the site  $q$  of  $\lambda$  becomes free. We then add  $q$  to  $Q$  with respect to its  $x$ -coordinate. If we do not hit any leader, we apply the same approach emanating a vertical half-line  $\ell_u$  from  $p$  upwards. Again, the first time  $\ell_u$  hits a right leader  $\lambda$  we apply the same rewiring obtaining a new leader  $\lambda'$ . The site  $q$  of  $\lambda$  is added to  $Q$  with respect to its  $x$ -coordinate. For an example see Fig 17b.

In both cases, the new leader stays on the horizontal segment of the old leader and on the segment of  $\ell_d$  or  $\ell_u$  that lies between  $p$  and  $\lambda$ . In particular, due to Lemma 10 the new leader stays below  $C_r$ . Thus, the solution remains crossing free. Further, the length of the horizontal segment of  $\lambda'$  is strictly smaller than the length of the horizontal segment of  $\lambda$ . The procedure therefore terminates. In particular, each site can only be added once to  $Q$  and finding the first leader that hits  $\ell_d$  or  $\ell_v$  takes  $O(n)$  time.

Thus, from  $\mathcal{L}_r$  we obtain a new solution  $\mathcal{L}'_r$  in  $O(n^2)$  time. We further can directly conclude the next lemma.

**Lemma 11.** *Let  $p$  be an arbitrary free site in  $B_r$  with respect to  $\mathcal{L}'_r$ , then the vertical line through  $p$  does not intersect any right leader of  $\mathcal{L}'_r$ .*

Note that all changes are applied in  $B_r$ , so Lemma 10 is still true for  $\mathcal{L}'_r$ . We can now show that the free sites in  $I_r$  are reachable from the free ports in  $I_r$ .

**Lemma 12.** *All free sites in  $I_r$  with respect to  $\mathcal{L}'_r$  are reachable from the free ports in  $I_r$  with respect to  $\mathcal{L}'_r$ .*

*Proof.* Assume that there is a free port  $x$  and a free site  $p$  in  $I_r$  such that  $\lambda(p, x)$  has a crossing with a leader  $\lambda^*$  of  $\mathcal{L}'_r$ . As already mentioned above, all sites below  $B_r$  are reachable from free ports of  $I_r$  so that  $p$  must lie in  $B_r$ .

Further, by Lemma 10 we know that  $x$  lies below  $B_r$  and therefore below the site of  $\lambda^*$ . Consequently, only the vertical segment of  $\lambda(p, x)$  can intersect  $\lambda^*$ . Thus, the vertical line through  $p$  intersects  $\lambda^*$ , which contradicts Lemma 11.  $\square$

Using a symmetric version of `MakeReachable`, we apply the same procedure on  $\mathcal{L}_\ell$  and the free sites in  $B_\ell$ , where  $B_\ell$  is the bounding box of  $\mathcal{L}_\ell$  analogously defined as  $B_r$ . Thus, we obtain the solution  $\mathcal{L}'_\ell$ , for which we can show an analogous statement.

**Lemma 13.** *All free sites in  $I_\ell$  with respect to  $\mathcal{L}'_\ell$  are reachable from the free ports in  $I_\ell$  with respect to  $\mathcal{L}'_\ell$ .*

We now solve the instance  $I_r$  restricted to free ports and sites with respect to  $\mathcal{L}'_r$  by means of the algorithm for the one-sided case as presented by Bekos et al. [4]; see Figure 18a. We denote the union of this solution and  $\mathcal{L}'_r$  by  $\mathcal{L}''_r$ . Due to Lemma 12, we know that  $\mathcal{L}''_r$  is free from any crossings, however, due to assumption A1, there are still some sites free with respect to  $\mathcal{L}''_r$ . We therefore apply `MakeReachable` on  $\mathcal{L}''_r$  and the remaining free sites of  $I_r$ , obtaining the solution  $\mathcal{L}'''_r$ ; see Fig. 18b. Note that Lemma 11 is also true for  $\mathcal{L}'''_r$  and the free sites of  $I_r$  with respect to  $\mathcal{L}'''_r$ .

**Lemma 14.** *All free sites in  $I_r \cup I_\ell$  with respect to  $\mathcal{L}'''_r \cup \mathcal{L}'_\ell$  are reachable from the free ports in  $I_\ell$  with respect to  $\mathcal{L}'_\ell$ .*

*Proof.* Assume that there is a free port  $x$  in  $I_\ell$  with respect to  $\mathcal{L}'_\ell$  and a free site  $p$  in  $I_r \cup I_\ell$  with respect to  $\mathcal{L}'''_r \cup \mathcal{L}'_\ell$  such that the leader  $\lambda(p, x)$  intersects a leader  $\lambda^*$  of  $\mathcal{L}'_\ell$  or  $\mathcal{L}'''_r$ . By Lemma 13 we know that  $\lambda^*$  cannot belong to  $\mathcal{L}'_\ell$ . But then  $\lambda^*$  belongs to  $\mathcal{L}'''_r$ . As  $\lambda^*$  and  $\lambda(x, p)$  are of opposite types, an intersection of  $\lambda(p, x)$  and  $\lambda^*$  means that  $p$  lies above or below the horizontal segment of  $\lambda^*$ , i.e., the vertical line through  $p$  intersects  $\lambda^*$ . This is a contradiction to Lemma 11, which also holds for  $\mathcal{L}'''_r$ .  $\square$

Due to Lemma 13 and Lemma 14 we can apply the algorithm for the one-sided case as described by Bekos et al. [4] on the remaining free ports and sites. Thus, finally we have obtained a crossing free solution for  $I_\ell \cup I_r$ . As there is always a solution for the one-sided case, the only possibility for the

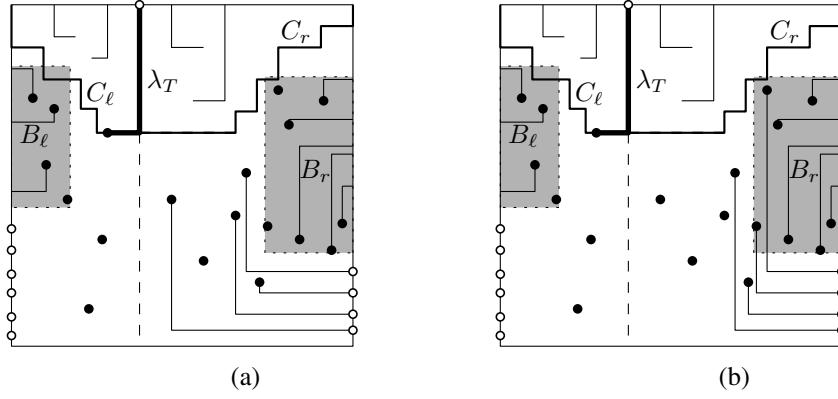


Fig. 18: a) The instance of Fig. 17b after solving the one-sided labeling for the right free ports. b) The same instance after applying MakeReachable.

algorithm to return no solution is when the dynamic program does not yield a solution. However, then by Lemma 9 there does not exist any solution for the choice of  $\lambda_T$ . Due to the previous lemmas and reasoning the algorithm is therefore correct. Further, considering a single leader  $\lambda_T$ , we need  $O(n^2)$  time. Thus, exploring all  $O(n^2)$  choices of  $\lambda_T$ , we need  $O(n^4)$  time all together.

**Theorem 7.** THREE-SIDED BOUNDARY LABELING can be solved in  $O(n^4)$  time using  $O(n)$  space.

## 6 Conclusion

In this paper, we have studied the problem of testing whether an instance of TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES admits a planar solution. We have given the first efficient algorithm for this problem, running in  $O(n^2)$  time.

The presented algorithm can also be used to solve a variety of different extensions of the problem. We have shown how to generalize to sliding ports instead of fixed ports without increasing the running time and how to maximize the number of labeled sites such that the solution is planar in  $O(n^3 \log n)$  time. We further have given an extension to the algorithm that minimizes the total leader length in  $O(n^8 \log n)$  time.

With some additional work, the presented approach can also be used to solve THREE-SIDED and FOUR-SIDED BOUNDARY LABELING in polynomial time. We have introduced an algorithm solving the three-sided case in  $O(n^4)$  time and the four-sided case in  $O(n^{10})$  time. Also, except for the length minimization, all presented extensions carry over. It remains open whether a minimum length solution of THREE-SIDED and FOUR-SIDED BOUNDARY LABELING can be computed in polynomial time.

## References

- [1] P. K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.*, 29(3):912–953, 1999. 3
- [2] M. A. Bekos, M. Kaufmann, M. Nöllenburg, and A. Symvonis. Boundary labeling with octilinear leaders. *Algorithmica*, 57(3):436–461, 2010. 2
- [3] M. A. Bekos, M. Kaufmann, K. Potika, and A. Symvonis. Area-feature boundary labeling. *Comput. J.*, 53(6):827–841, 2010. 2
- [4] M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Comput. Geom. Theory Appl.*, 36(3):215–236, 2007. 1, 2, 21

- [5] M. Benkert, H. J. Haverkort, M. Kroll, and M. Nöllenburg. Algorithms for multi-criteria boundary labeling. *J. Graph Algorithms Appl.*, 13(3):289–317, 2009. 2, 4
- [6] B. Chazelle and 36 co-authors. The computational geometry impact task force report. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223, pages 407–463. American Mathematical Society, Providence, RI, 1999. 1
- [7] M. Fink, J.-H. Haunert, A. Schulz, J. Spoerhase, and A. Wolff. Algorithms for labeling focus regions. *IEEE Trans. Visual. Comput. Graphics*, 18(12):2583–2592, 2012. 2
- [8] H. Freeman, S. Marrinan, and H. Chitalia. Automated labeling of soil survey maps. In *Proc. ASPRS-ACSM Annual Convention, Baltimore*, volume 1, pages 51–59, 1996. 1
- [9] A. Gemsa, J.-H. Haunert, and M. Nöllenburg. Boundary-labeling algorithms for panorama images. In *Proc. 19th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inform. Syst.*, pages 289–298, 2011. 2
- [10] P. Gritzmann, B. Mohar, J. Pach, and R. Pollack. Embedding a planar triangulation with vertices at specified positions. *Amer. Math. Mon.*, 98:165–166, 1991. 3
- [11] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Comm. ACM*, 18(6):341–343, 1975. 9, 10
- [12] B. Katz, M. Krug, I. Rutter, and A. Wolff. Manhattan-geodesic embedding of planar graphs. In D. Eppstein and E. R. Gansner, editors, *Proc. 17th Int. Symp. Graph Drawing (GD’09)*, volume 5849 of *Lecture Notes Comput. Sci.*, pages 207–218. Springer-Verlag, 2010. 3
- [13] P. Kindermann, B. Niedermann, I. Rutter, M. Schaefer, A. Schulz, and A. Wolff. Two-sided boundary labeling with adjacent sides. Arxiv report, Apr. 2013. Available at <http://arxiv.org/abs/TODO>. 4
- [14] J. L. Morrison. Computer technology and cartographic change. In D. Taylor, editor, *The Computer in Contemporary Cartography*. Johns Hopkins University Press, 1980. 1
- [15] M. Nöllenburg, V. Polishchuk, and M. Sysikaski. Dynamic one-sided boundary labeling. In *Proc. 18th Internat. ACM Symp. Adv. Geogr. Inform. Syst. 10*, pages 310–319, 2010. 2
- [16] R. Raghavan, J. Cohoon, and S. Sahni. Single bend wiring. *J. Algorithms*, 7(2):232–257, 1986. 3
- [17] B. Speckmann and K. Verbeek. Homotopic rectilinear routing with few links and thick edges. In A. López-Ortiz, editor, *Proc. LATIN 2010*, volume 6034 of *Lecture Notes Comput. Sci.*, pages 468–479. Springer, 2010. 3
- [18] M. van Kreveld, T. Strijk, and A. Wolff. Point labeling with sliding labels. *Comput. Geom. Theory Appl.*, 13:21–47, 1999. 1
- [19] S. Zoraster. Practical results using simulated annealing for point feature label placement. *Cartography and GIS*, 24(4):228–238, 1997. 1